
IBM System p5 570

Model 9117-570

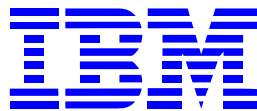
Using

AIX 5L Version 5.3

and

DB2 Universal Database 8.2

TPC BenchmarkTM C
Full Disclosure Report



First Edition

February 14, 2006

Special Notices

The following terms used in this publication are trademarks of **International Business Machines** Corporation in the United States and/or other countries:

IBM eServer pSeries

IBM eServer xSeries

AIX

IBM

DB2, DB2 UDB, DB2 Universal Database

The following terms used in this publication are trademarks of other companies as follows:

TPC Benchmark, TPC-C, and tpmC are trademarks of the Transaction Processing Performance Council

Microsoft Windows 2000 server and COM+ are registered trademarks of Microsoft Corporation

First Edition: February 14, 2006

The information contained in this document is distributed on an AS IS basis without any warranty either expressed or implied. The use of this information or the implementation of any of these techniques is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item has been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used.

It is possible that this material may contain references to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such products, programming, or services in your country.

All performance data contained in this publication was obtained in a controlled environment, and therefore the results which may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data in their specific environment.

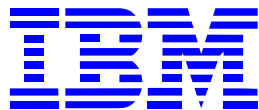
Request for additional copies of this document should be sent to the following address:

TPC Benchmark Administrator
IBM Commercial Performance
Mail Stop 9571
11501 Burnet Road
Austin, TX 78758
FAX Number (512) 838-1852

© Copyright International Business Machines Corporation, 2004. All rights reserved.

Permission is hereby granted to reproduce this document in whole or in part, provided the copyright notice printed above is set forth in full text on the title page of each item reproduced.

NOTE: US. Government Users - Documentation related to restricted rights: Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

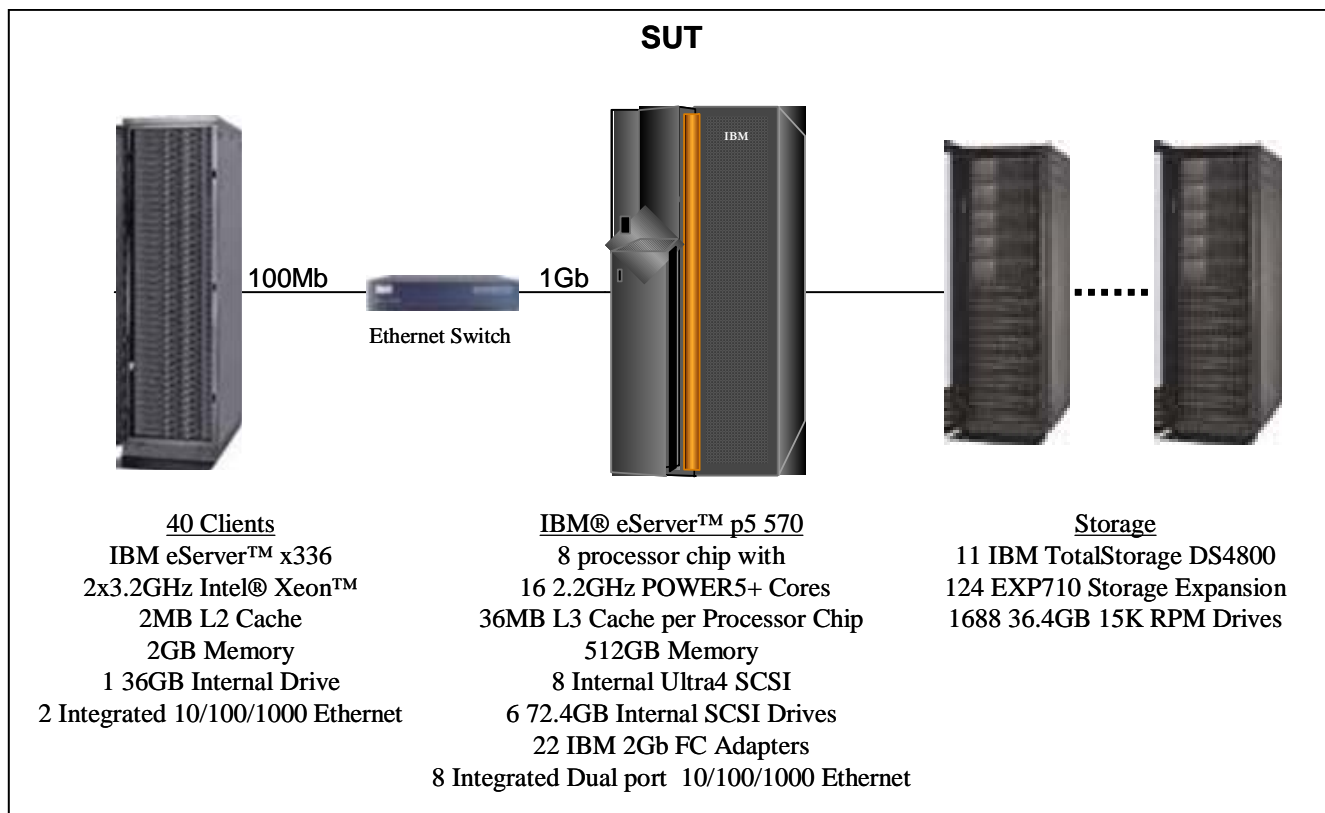


**IBM System p5 570
Model 9117-570**

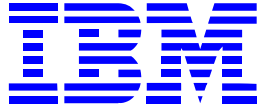
TPC-C Rev. 5.6

Report Date: February 14, 2006

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date	
\$4,528,937 USD	1,025,169.69	\$4.42 USD	May 31, 2006	
Database Server Processor Chip/Core/Thread	Database Manager	Operating System	Other Software	No. Users
8/16/32 POWER5+ 2.2GHz	DB2 UDB 8.2	AIX 5L V5.3	Microsoft Visual C++ Microsoft COM+	816,000



System Components	Each of the 40 Clients		Server	
	Quantity	Description	Quantity	Description
Processors Chip/Core/Threads	2/2/4	3.2GHz 1MB L3 Xeon Processor	8/16/32	2.2GHz POWER5+™
Memory	2	1024 MB	16	32 GB
Disk Controllers	1	Ultra320 SCSI	2 22 11	Integrated dual Ultra3SCSI 2Gb FC Adapters IBM DS4800 Controllers
Disk Drives	1	36GB	1688 6	36.4GB 15K RPM FC 72.4GB 15K RPM SCSI
Total Storage		1,476 GB		61,877.6GB
Terminals	1	System Console	1	System Console

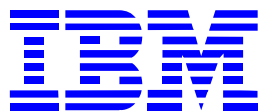


IBM System p5 570 Model 9117-570

TPC-C Rev. 5.6

Report Date: February 14, 2006

Description	Part No.	Source	Unit Price	Qty	Ext Price	Maint Price
Server Hardware						
Server 1:9117 Model 570	9117-570	1	3,867	1	3,867	7,146
Op Panel	1846	1	199	1	199	0
Processor Cable, Four-Drawer System	1849	1	17,868	1	17,868	0
SP Flex Cable, Four-Drawer System	1859	1	7,412	1	7,412	0
IDE Slimline DVD-ROM Drive	2640	1	274	1	274	0
73.4 GB 10,000 RPM Ultra320 SCSI Disk Drive	3274	1	599	6	3,594	0
32GB (4x8GB) DIMMS (533MHZ DELHI) DDR2	4498	1	175,000	16	2,800,000	0
IBM 10/100/1000 Base-TX Ethernet PCI-X Adapter	5701	1	699	0	0	0
2 Gigabit Fibre Channel PCI-X Adapter	5716	1	1,999	6	11,994	0
SPCN 2m Cable	6001	1	33	2	66	0
Power Cord (9-foot), Drawer to IBM PDU, 250V/10A	6671	1	19	8	152	0
IBM Rack-mount Drawer Rail Kit	7164	1	222	4	888	0
2-WAY 2.2GHZ GS DCM PROC, 36MB L3, 8 DDR2	8338	1	10,720	8	85,760	33,792
CEC Backplane	7865	1	1,588	4	6,352	0
I/O Backplane	7866	1	5,426	4	21,704	0
Midplane	7867	1	662	4	2,648	0
DASD Backplane	7868	1	1,588	4	6,352	0
Media Backplane	7869	1	185	1	185	0
Power Midplane	7870	1	265	4	1,060	0
PROCESSOR POWER REGULATOR	7768	1	675	12	8,100	0
Serial Port Riser Card	7878	1	132	4	528	0
System Drawer Enclosure	7879	1	463	4	1,852	0
AC Power Supply, 240V, 1400 W	7888	1	1,059	8	8,472	0
MODEL 570 PERMANENT PROCESSOR ACTIVATION FEATURE	7618	1	21,440	16	343,040	59,520
Service Processor Card (FSP)	7997	1	860	1	860	0
I/O Drawer 1:7311 Model D11	7311-D11	1	4,461	4	17,844	20,324
RIO-2 (Remote I/O-2) Cable, 3.5M	3147	1	728	4	2,912	0
2 Gigabit Fibre Channel PCI-X Adapter	5716	1	1,999	16	31,984	0
SPCN 3m Cable	6006	1	53	4	212	0
AC Power Supply, 288 W	6276	1	375	8	3,000	0
Rio-2 Remote I/O Loop Adapter	6438	1	900	4	3,600	0
Power Cord (12ft) 250V/10A, RA	6459	1	5	8	40	0
Planar Board, 6 PCI-X Slots	6582	1	500	4	2,000	0
Air Blower	6583	1	100	4	400	0
Dual I/O Unit Enclosure	7311	1	417	4	1,668	0
Rack Model T42	7014-T42	1	3,970	1	3,970	888
Front door (Black) for High Perforation (2m racks)	6069	1	550	1	550	0
Side Panel (Black)	6098	1	150	2	300	0
PDU to 14', 200-240V/24A, UTG0247, PT#12	6654	1	240	3	720	0
Power Dist Unit-Side Mount, Universal UTG0247	7188	1	1,200	2	2,400	0
HMC 1:7310-C04 Desktop Hardw.Mgmt.Console	7310-C04	1	1,830	1	1,830	1,344
IBM ThinkVision C170 17-inch Color Monitor	3631	1	250	1	250	0
Power Cord (6-foot), To Wall Plug Type #4	6470	1	14	2	28	0
Ethernet Cable, 6M, HMC to System Unit	7801	1	12	1	12	0
Keyboard - English, #103P	8800	1	83	1	83	0
Mouse - Attachment Cable	8841	1	62	1	62	0
			Subtotal		3,407,092	123,014



**IBM System p5 570
Model 9117-570**

TPC-C Rev. 5.6

Report Date: February 14, 2006

Storage

DS4000 EXP710 Storage Expansion	1740-710	1	6,000	124	744,000	94,240
(19K1271) 2Gb Fibre Channel Short Wave GBIC	2210	1	499	518	258,482	0
(06P5772) 2Gb FC, 36.4GB/15K Drive	5212	1	1,115	1,688	1,882,120	0
(19K1247) 1m 50u Fiber Optic Cable (LC-LC)	5601	1	79	248	19,592	0
DS4800 Disk System Model 82 (4 GB Cache)	1815-82A	1	53,995	11	593,945	35,200
(22R4255) DS4800 AIX Host Kit	7711	1	7,000	1	7,000	0
DS4800 8-Storage Partitions	8870	1	10,000	11	110,000	0
(19K1248) Fiber Cable 5m Multimode (LC-LC)	5605	1	129	22	2,838	0
			Subtotal		3,617,977	129,440

Server Software

AIX 5.3 (media only)	5692-A5L	1	50	1	50	0
AIX 5L V5.3	5765-G03	1	NC	1	0	0
Per Processor F5 AIX 5L V5.3	0005	1	1,225	16	19,600	
Software Maintenance for AIX, 3 Year	5773-SM3	1	NC	1		
F5 3 Yr SWMA for AIX per Processor Reg/Ren	0466	1	1,958	16		31,328
F5 3 yr Services 7x24 Support per Processor	0468	1	496	16	0	7,936
HMC Initial Software Support 3 Year	5773-RS3	1	NC	1		
Per Processor Software Support 3 Year	569	1	675	1		675
Per Processor 24x7 Software Support 3 Year	570	1	236	1	0	236
C for AIX user Lic+SW maint 12 MO	D5A1DLL	1	515	1	515	0
C for AIX user annual SW maint renewal	E1A1FLL	1	103	2	0	206
DB2 Enterprise Server Edition Proc Lic/1 yr Maint.		1	23,902	16	382,432	0
DB2 Enterprise Server Ed Proc Maint Renew	E00BILL	1	1,138	32		36,416
			Subtotal		402,597	76,797

Client Hardware and Software

xSeries 336	8837E3U	1	3,319	40	132,760	18,000
36GB 15K U320 SCSI HS Express	30R5096	1	269	40	10,760	0
NetBay42 Standard Rack	93074SX	1	1,489	11	16,379	0
Optical 3-Button Mouse - USB	90P0744	1	15	1	15	
Preferred Pro Full Size PS/2 Keyboard	25R6968	1	29	1	29	0
IBM C117 17" CRT Monitor	49387NU	1	149	1	149	0
			Subtotal		160,092	18,000

Third Party Hardware/Software

Visual C++ Standard Edition	254-00170	2	109	1	109	0
Microsoft Windows 2000 Server	C11-00821	2	738	40	4,428	
Microsoft Problem Resolution Services		2	245	1		245
NETGEAR GS116 16-port 10/100/1000Mbps Gigabit Switch	638864	3	200	6	1,200	0
			Subtotal		5,737	245

Total 7,621,279 346,608

IBM Total System Discounts* -3,423,619

Three-Year Cost of Ownership 4,544,268

Notes:

For pricing details and contact information please see appendix E

Pricing Sources: 1) IBM 2) Microsoft 3) CDW

tpmC 1,025,170

\$/tpmC 4.43

*Discounts are based on US list prices for similar quantities & configurations including pre-payment for maintenance. The discount applies to the totality of all items with price source of "1".

Audited by: Francois Raab, Info Sizing (www.infosizing.com)

Prices used in TPC benchmarks reflect the actual prices a customer would pay for a one-time purchase of the stated components. Individually negotiated discounts are not permitted. Special prices based on assumptions about past or future purchases are not permitted. All discounts reflect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark specifications. If you find that the stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank you

Numerical Quantities Summary for the IBM System p5 570 Model 9117-570

MQTH, computed Maximum Qualified Throughput: 1,025,169.69 tpmC

<u>Response Times (in seconds)</u>	<u>90th %</u>	<u>Average</u>	<u>Maximum</u>
New Order	0.50	0.31	3.23
Payment	0.50	0.31	3.45
Order-Status	0.50	0.32	3.21
Delivery (interactive)	0.25	0.17	2.84
Delivery (deferred)	0.80	0.40	3.70
Stock-Level	0.65	0.40	3.55
Menu	0.23	0.14	2.85

Response time delay added for emulated components was 0.1 seconds

<u>Transaction Mix, in percent of total transactions</u>	<u>Percent</u>
New Order	44.91%
Payment	43.03%
Order-Status	4.02%
Delivery	4.01%
Stock-Level	4.01%

<u>Keying/Think Times (in seconds)</u>	<u>Min.</u>	<u>Average</u>	<u>Max.</u>
New Order	18.00/0.01	18.01/12.02	18.05/120.20
Payment	3.00/0.01	3.01/12.02	3.05/120.21
Order-Status	2.00/0.01	2.01/10.01	2.04/100.10
Delivery	2.00/0.01	2.01/5.02	2.05/50.20
Stock-Level	2.00/0.01	2.01/5.02	2.05/50.20

Test Duration

Ramp-up Time	29 minutes
Measurement interval	2 hours 30 minutes
Transactions during measurement interval (all types)	342,405,437
Ramp-down time	20 minutes

Checkpoints

Number of checkpoints	N/A
Checkpoint interval	N/A

Table of Content

Preface	10
0 General Items.....	11
0.1. Application Code Disclosure.....	11
0.2. Benchmark Sponsor.....	11
0.3. Parameter Settings.....	11
0.4. Configuration Diagrams	11
1 Clause 1: Logical Data Base Design Related Items.....	13
1.1. Table Definitions	13
1.2. Database Organization.....	13
1.3. Insert and/or Delete Operations	13
1.4. Horizontal or Vertical Partitioning	13
2 Clause 2: Transaction & Terminal Profiles Related Items.....	14
2.1. Verification for the Random Number Generator.....	14
2.2. Input/Output Screens	14
2.3. Priced Terminal Features.....	14
2.4. Presentation Managers.....	14
2.5. Home and Remote Order-lines	14
2.6. New-Order Rollback Transactions	14
2.7. Number of Items per Order	15
2.8. Home and Remote Payment Transactions	15
2.9. Non-Primary Key Transactions	15
2.10. Skipped Delivery Transactions.....	15
2.11. Mix of Transaction Types.....	16
2.12. Queuing Mechanism of Delivery.....	16
3 Clause 3: Transaction and System Properties	17
3.1. Atomicity Requirements.....	17
3.2. Consistency Requirements	17
3.3. Isolation Requirements.....	18
3.4. Durability Requirements.....	18
4 Clause 4: Scaling and Data Base Population Related Items	20
4.1. Cardinality of Tables	20
4.2. Distribution of Tables and Logs	20
4.3. Data Base Model Implemented	20
4.4. Partitions/Replications Mapping	21
4.5. 60-Day Space Calculations.....	24
5 Clause 5: Performance Metrics and Response Time Related Items.....	25
5.1. Response Times.....	25
5.2. Keying and Think Times	25
5.3. Response Time Frequency Distribution	26
5.4. Performance Curve for Response Time versus Throughput.....	28
5.5. Think Time Frequency Distribution	29
5.6. Throughput versus Elapsed Time	30
5.7. Steady State Determination	30
5.8. Work Performed During Steady State	30
5.9. Measurement Interval.....	32
6 Clause 6: SUT, Driver, and Communication Definition Related Items.....	33
6.1. RTE Availability	33
6.2. Functionality and Performance of Emulated Components	33
6.3. Network Bandwidth.....	33
6.4. Operator Intervention	33
7 Clause 7: Pricing Related Items.....	34
7.1. Hardware and Programs Used	34
7.2. Three Year Cost of System Configuration	34
7.3. Availability Dates.....	34
7.4. Statement of tpmC and Price/Performance.....	34

8	Clause 9: Audit Related Items	35
	Appendix - A: Client Server Code	38
	A.1 Client/Terminal Handler Code	38
	A.2 Client Transaction Code	52
	Appendix - B: Tunable Parameters	98
	B.1 Database Parameters	98
	B.2 Transaction Monitor Parameters	99
	B.3 AIX Parameters	103
	Appendix - C: Database Setup Code	106
	C.1 Database Creation Scripts	106
	C.2 Data Generation Code	313
	Appendix - D: RTE Scripts	328
	D.1 RTE Parameters	328
	D.2 RTE Scripts	333
	Appendix - E: Third Party Pricing Information	344

Abstract

This report documents the full disclosure information required by the TPC Benchmark™ C Standard Specification Revision 5.6 dated April, 2004, for measurements on the IBM System p5 570 Model 9117-570. The software used on the IBM System p5 570 Model 9117-570 includes AIX 5L Version 5.3 operating system, DB2 UDB 8.2 database manager. Microsoft COM+ is used as transaction manager.

IBM System p5 570 Model 9117-570

Company Name	System Name	Data Base Software	Operating System Software
IBM Corporation	IBM System p5 570 Model 9117-570	DB2 UDB 8.2	AIX 5L Version 5.3

Total System Cost	TPC-C Throughput	Price/Performance
<ul style="list-style-type: none">• Hardware• Software• 3 Years Maintenance	Sustained maximum throughput of system running TPC-C expressed in transactions per minute	Total system cost/tpmC
\$4,528,937 USD	1,025,169.69	\$4.42 USD

Preface

TPC Benchmark™ C Standard Specification was developed by the Transaction Processing Performance Council (TPC). It was released on August 13, 1992 and updated with revision 5.6 in October 2005.

This is the full disclosure report for benchmark testing of the IBM System p5 570 Model 9117-570 and DB2 UDB 8.2 according to the TPC Benchmark™ C Standard Specification.

TPC Benchmark™ C exercises the system components necessary to perform tasks associated with that class of on-line transaction processing (OLTP) environments emphasizing a mixture of read-only and update intensive transactions. This is a complex OLTP application environment exercising a breadth of system components associated by such environments characterized by:

- The simultaneous execution of multiple transaction types that span a breadth of complexity
- On-line and deferred transaction execution modes
- Multiple on-line terminal sessions
- Moderate system and application execution time
- Significant disk input/output
- Transaction integrity (ACID properties)
- Non-uniform distribution of data access through primary and secondary keys
- Data bases consisting of many tables with a wide variety of sizes, attributes, and relationships
- Contention on data access and update

This benchmark defines four on-line transactions and one deferred transaction, intended to emulate functions that are common to many OLTP applications. However, this benchmark does not reflect the entire range of OLTP requirements. The extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-C approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to any other environment are not recommended.

Benchmark results are highly dependent upon workload, specific application requirements, and systems design and implementation. Relative system performance will vary as a result of these and other factors. Therefore, TPC-C should not be used as a substitute for a specific customer application benchmarks when critical capacity planning and/or product evaluation decisions are contemplated.

The performance metric reported by TPC-C is a “business throughput” measuring the number of orders processed per minute. Multiple transactions are used to simulate the business activity of processing an order, and each transaction is subject to a response time constraint. The performance metric for this benchmark is expressed in transactions-per-minute-C (tpmC). To be compliant with the TPC-C standard, all references to tpmC results must include the tpmC rate, the associated price-per-tpmC, and the availability date of the priced configuration.

0 General Items

0.1. Application Code Disclosure

The application program (as defined in Clause 2.1.7) must be disclosed. This includes, but is not limited to, the code implementing the five transactions and the terminal input and output functions.

Appendix A contains the IBM pSeries application code for the five TPC Benchmark™ C transactions. Appendix D contains the terminal functions and layouts.

0.2. Benchmark Sponsor

A statement identifying the benchmark sponsor(s) and other participating companies must be provided.

This benchmark was sponsored by **International Business Machines Corporation.**

0.3. Parameter Settings

Settings must be provided for all customer-tunable parameters and options which have been changed from the defaults found in actual products, including but not limited to:

- *Data Base tuning options*
- *Recovery/commit options*
- *Consistency/locking options*
- *Operating system and application configuration parameters.*

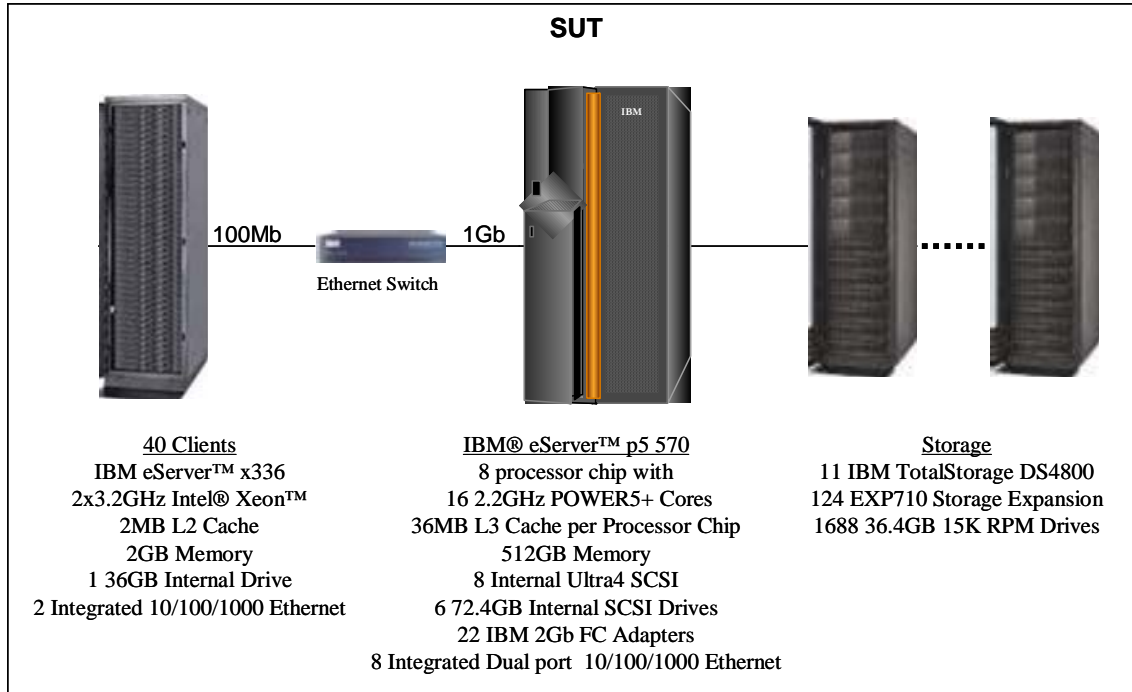
Appendix B contains the system, data base, and application parameters changed from their default values used in these TPC Benchmark™ C tests.

0.4. Configuration Diagrams

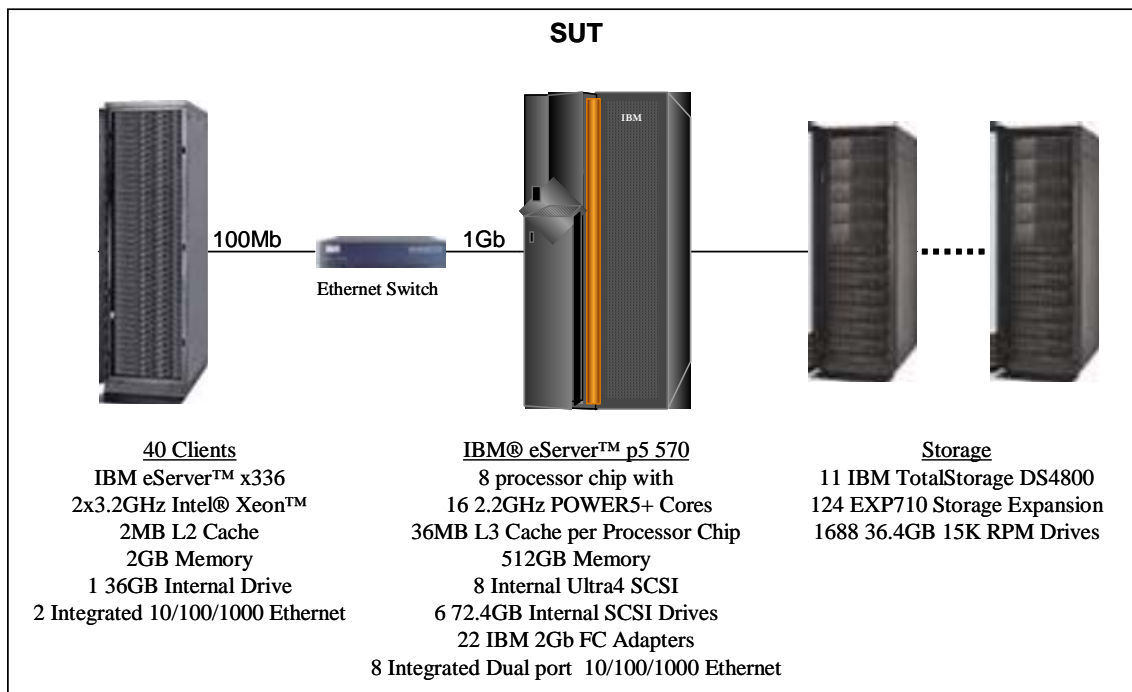
Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:

- *Number and type of processors*
- *Size of allocated memory, and any specific mapping/partitioning of memory unique to the test*
- *Number and type of disk units (and controllers, if applicable)*
- *Number of channels or bus connections to disk units, including the protocol type*
- *Number of LAN (e.g. Ethernet) connections, including routers, work stations, terminals, etc, that were physically used in the test or are incorporated into the pricing structure (see Clause 8.1.8)*
- *Type and run-time execution location of software components (e.g. DBMS, client processes, transaction monitors, software drivers, etc)*

IBM System p5 570 Model 9117-570 Benchmark Configuration



IBM System p5 570 Model 9117-570 Priced Configuration



1 Clause 1: Logical Data Base Design Related Items

1.1. Table Definitions

Listings must be provided for all table definition statements and all other statements used to setup the data base.

Appendix C contains the table definitions and the database load programs used to build the data base.

1.2. Database Organization

The physical organization of tables and indices, within the data base, must be disclosed.

Physical space was allocated to DB2 UDB on the server disks according to the details provided in Appendix C.

1.3. Insert and/or Delete Operations

It must be ascertained that insert and/or delete operations to any of the tables can occur concurrently with the TPC-C transaction mix. Furthermore, any restriction in the SUT data base implementation that precludes inserts beyond the limits defined in Clause 1.4.11 must be disclosed. This includes the maximum number of rows that can be inserted and the maximum key value for these new rows.

There were no restrictions on insert and/or delete operations to any of the tables. The space required for an additional five percent of the initial table cardinality was allocated to DB2 UDB and priced as static space.

The insert and delete functions were verified by the auditor. In addition, the auditor verified that the primary key for each database table could be updated outside the range of its initial partition.

1.4. Horizontal or Vertical Partitioning

While there are few restrictions placed upon horizontal or vertical partitioning of tables and rows in the TPC-C benchmark, any such partitioning must be disclosed.

All tables but ITEM were horizontally partitioned into multiple tables.

Each table partition for STOCK, CUSTOMER, ORDERS, ORDERLINE, WAREHOUSE, DISTRICT, NEWORDER, and HISTORY contains data associated with a range of 1,600 warehouses.

For each partitioned table, a view was created over all table partitions to provide full transparency of data manipulation.

No tables were replicated.

2 Clause 2: Transaction & Terminal Profiles Related Items

2.1. Verification for the Random Number Generator

The method of verification for the random number generation must be disclosed.

The `srandom()`, `getpid()` and `gettimeofday()` functions are used to produce unique random seeds for each driver. The drivers use these seeds to seed the `srand()`, `srandom()` and `srand48()` functions. Random numbers are produced using wrappers around the standard system random number generators.

The negative exponential distribution uses the following function to generate the distribution. This function has the property of producing a negative exponential curve with a specified average and a maximum value 4 times the average.

```
const double RANDOM_4_Z = 0.89837799236185
const double RANDOM_4_K = 0.97249842407114
double neg_exp_4(double average {
    return - average * (1/RANDOM_4_Z * log (1 - RANDOM_4_K * drand48()));
})
```

The seeds for each user were captured and verified by the auditor to be unique. In addition, the contents of the database were systematically searched and randomly sampled by the auditor for patterns that would indicate the random number generator had affected any kind of a discernible pattern; none were found.

2.2. Input/Output Screens

The actual layouts of the terminal input/output screens must be disclosed.

The screen layouts are now presented in HTML 1.0 web pages. Clauses 2.4.3, 2.5.3, 2.6.3, 2.7.3, and 2.8.3 of the TPC-C specifications were used as guidelines for html character placement.

2.3. Priced Terminal Features

The method used to verify that the emulated terminals provide all the features described in Clause 2.2.2.4 must be explained. Although not specifically priced, the type and model of the terminals used for the demonstration in 8.1.3.3 must be disclosed and commercially available (including supporting software and maintenance).

The emulated workstations, IBM xSeries 336 systems, are commercially available and support all of the requirements in Clause 2.2.2.4.

2.4. Presentation Managers

Any usage of presentation managers or intelligent terminals must be explained.

The workstations did not involve screen presentations, message bundling or local storage of TPC-C rows. All screen processing was handled by the client system. All data manipulation was handled by the server system.

2.5. Home and Remote Order-lines

The percentage of home and remote order-lines in the New-Order transactions must be disclosed.

Table 2-1 shows the percentage of home and remote transactions that occurred during the measurement period for the New-Order transactions.

2.6. New-Order Rollback Transactions

The percentage of New-Order transactions that were rolled back as a result of an illegal item number must be disclosed.

Table 2-1 shows the percentage of New-Order transactions that were rolled back due to an illegal item being entered.

2.7. Number of Items per Order

The number of items per order entered by New-Order transactions must be disclosed.

Table 2-1 show the average number of items ordered per New-Order transaction.

2.8. Home and Remote Payment Transactions

The percentage of home and remote Payment transactions must be disclosed.

Table 2-1 shows the percentage of home and remote transactions that occurred during the measurement period for the Payment transactions.

2.9. Non-Primary Key Transactions

The percentage of Payment and Order-Status transactions that used non-primary key (C_LAST) access to the data base must be disclosed.

Table 2-1 shows the percentage of non-primary key accesses to the data base by the Payment and Order-Status transactions.

2.10. Skipped Delivery Transactions

The percentage of Delivery transactions that were skipped as a result of an insufficient number of rows in the NEW-ORDER table must be disclosed.

Table 2-1 shows the percentage of Delivery transactions missed due to a shortage of supply of rows in the NEW-ORDER table.

2.11. Mix of Transaction Types

The mix (i.e. percentages) of transaction types seen by the SUT must be disclosed.

Table 2-1 shows the mix percentage for each of the transaction types executed by the SUT.

New Order	IBM System p5 570 Model 9117-570
Percentage of Home order lines	99.01%
Percentage of Remote order lines	0.99%
Percentage of Rolled Back Transactions	1.00%
Average Number of Items per order	10.00
Payment	
Percentage of Home transactions	85.00%
Percentage of Remote transactions	15.00%
Non-Primary Key Access	
Percentage of Payment using C_LAST	60.00%
Percentage of Order-Status using C_LAST	59.97%
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.91%
Payment	43.03%
Order-Status	4.02%
Delivery	4.01%
Stock-Level	4.01%

Table 2-1: Numerical Quantities for Transaction and Terminal Profiles

2.12. Queuing Mechanism of Delivery

The queuing mechanism used to defer execution of the Delivery transaction must be disclosed.

The Delivery transaction was submitted to an ISAPI queue that is separate from the COM+ queue that the other transactions used. This queue is serviced by a variable amount of threads that are separate from the worker threads inside the web server. Web server threads are able to complete the on-line part of the Delivery transaction and immediately return successful queuing responses to the drivers. The threads servicing the queue are responsible for completing the deferred part of the transaction asynchronously.

3 Clause 3: Transaction and System Properties

The results of the ACID test must be disclosed along with a description of how the ACID requirements were met.

All ACID tests were conducted according to specification.

3.1. Atomicity Requirements

The system under test must guarantee that data base transactions are atomic; the system will either perform all individual operations on the data, or will assure that no partially-completed operations leave any effects on the data.

3.1.1. Atomicity of Completed Transaction

Perform the Payment transaction for a randomly selected warehouse, district, and customer (by customer number) and verify that the records in the CUSTOMER, DISTRICT, and WAREHOUSE tables have been changed appropriately.

The following steps were performed to verify the Atomicity of completed transactions.

1. The balance, BALANCE_1, was retrieved from the CUSTOMER table for a random Customer, District and Warehouse combination.
2. The Payment transaction was executed and committed for the Customer, District, and Warehouse combination used in step 1.
3. The balance, BALANCE_2, was retrieved again for the Customer, District, and Warehouse combination used in step 1 and step 2. It was verified that BALANCE_1 was greater than BALANCE_2 by the amount of the Payment transaction.

3.1.2. Atomicity of Aborted Transactions

Perform the Payment transaction for a randomly selected warehouse, district, and customer (by customer number) and substitute a ROLLBACK of the transaction for the COMMIT of the transaction. Verify that the records in the CUSTOMER, DISTRICT, and WAREHOUSE tables have NOT been changed.

The following steps were performed to verify the Atomicity of the aborted Payment transaction:

1. The Payment application code was implemented with a Perl script that allowed the transaction to be rolled back rather than committed.
2. The balance, BALANCE_3, was retrieved from the Customer table for the same Customer, District, and Warehouse combination used in the completed Payment transaction Atomicity test.
3. The Payment transaction was executed for the Customer, District and Warehouse used in step 2. Rather than commit the transaction, the transaction was rolled back.
4. The balance, BALANCE_4 was retrieved again for the Customer, District, and Warehouse combination used in step 2. It was verified that BALANCE_4 was equal to BALANCE_3, demonstrating that there were no remaining effects of the rolled back Payment transaction.

3.2. Consistency Requirements

Consistency is the property of the application that requires any execution of a data base transaction to take the data base from one consistent state to another, assuming that the data base is initially in a consistent state.

Verify that the data base is initially consistent by verifying that it meets the consistency conditions defined in Clauses 3.3.2.1 to 3.3.2.4. Describe the steps used to do this in sufficient detail so that the steps are independently repeatable.

The specification defines 12 consistency conditions of which the following four are required to be explicitly demonstrated:

1. The sum of balances (d_ytd) for all Districts within a specific Warehouse is equal to the balance (w_ytd) of that Warehouse.
2. For each District within a Warehouse, the next available Order ID (d_next_o_id) minus one is equal to the most recent Order ID [max(o_id)] for the Order table associated with the preceding District and Warehouse.

Additionally, that same relationship exists for the most recent Order ID [max(o_id)] for the New Order table associated with the same District and Warehouse. Those relationships can be illustrated as follows:

$$d_next_o_id - 1 = \max(o_id) = \max(no_o_id)$$

where (d_w_id = o_w_id = no_w_id) and (d_id = o_d_id = no_d_id)

3. For each District within a Warehouse, the value of the most recent Order ID [max(no_o_id)] minus the first Order ID [min(no_o_id)] plus one, for the New Order table associated with the District and Warehouse equals the number of rows in that New Order table. That relationship can be illustrated as follows:

$$\max(no_o_id) - \min(no_o_id) + 1 = \text{number of rows in New Order for the Warehouse/District}$$

4. For each District within a Warehouse, the sum of Order Line counts [sum(o_ol_cnt)] for the Order table associated with the District equals the number of rows in the Order Line table associated with the same District. That relationship can be illustrated as follows:

$$\text{sum}(o_ol_cnt) = \text{number of rows in the Order Line table for the Warehouse/District}$$

An RTE driven run was executed against a freshly loaded database. After the run the 4 consistency conditions defined above were tested using a script to issue queries to the database. All queries showed that the database was still in a consistent state.

3.3. Isolation Requirements

Operations of concurrent data base transactions must yield results which are indistinguishable from the results which would be obtained by forcing each transaction to be serially executed to completion in some order.

The benchmark specification defines nine tests to demonstrate the property of transaction isolation. The tests, described in Clauses 3.4.2.1 – 3.4.2.9 were all successfully executed using a series of scripts. Case A was observed during the execution of Isolation Tests 7-9.

3.4. Durability Requirements

The tested system must guarantee durability: the ability to preserve the effects of committed transactions and insure data base consistency after recovery from any one of the failures listed in Clause 3.5.3

3.4.1. Permanent Unrecoverable Failure of any Single Durable Medium

Permanent irrecoverable failure of any single durable medium containing TPC-C data base tables or recovery log data.

Failure of Log Disk and Log Cache:

This test was conducted on a fully scaled database. The following steps were performed successfully.

1. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving SUM_1.
2. A full load test was started and allowed to run for over 10 minutes.
3. One of the disks containing the transaction log was removed. Since the disk was RAID-5, DB2 continued to process the transactions successfully.
4. The test continued for at least another 5 minutes.
5. Since write cache mirroring was enabled for the log device, one of the Fibre Channel controllers, which holds one copy of the mirrored cache, was removed. There was a brief pause in I/O while the failover to the remaining log controller occurred. The controller detected a mirror-out-of-sync condition and deactivated write-back cache.
6. The run continued without write-back cache.
7. The system was subsequently shut down.

8. The disk from step 3 was replaced.
9. The system was powered back on and DB2 was allowed to recover.
10. Step 1 was performed returning the value for SUM_2. It was verified that SUM_2 was greater than SUM_1 plus the completed New_Order transactions recorded by the RTE. The additional transactions found in the database were attributed to in-flight activity at the time of the failure.

Failure of Durable Medium Containing TPC-C Database Tables:

The following steps were successfully performed to demonstrate Durability against the failure of a disk unit with database tables:

1. The contents of the database were backed up in full.
2. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving SUM_1.
3. A scaled-down test was started with 25% of the full load.
4. A disk containing the TPCC table was removed causing DB2 to report numerous errors when attempting to access that device
5. The disk was powered back on and the full database was restored from the backup copy in step 1.
6. DB2 was restarted and the transactions in the log were applied to the database.
7. Step 2 was performed returning SUM_2. It was verified that SUM_2 was greater than SUM_1 plus the completed New_Order transactions recorded by the RTE. The additional transactions found in the database were attributed to in-flight activity at the time of the failure.
8. Consistency condition 3 was verified.

Instantaneous Interruption and Memory Failure:

The following steps were conducted on a fully scaled database:

1. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving SUM_1.
2. A full load test was started and allowed to run for over 10 minutes.
3. The system was powered off, which removed power from all system components, including memory.
4. The system was powered back on and DB2 recovered.
5. Step 1 was performed returning SUM_2. It was verified that SUM_2 was greater than SUM_1 plus the completed New_Order transactions recorded by the RTE. The additional transactions found in the database were attributed to in-flight activity at the time of the failure
6. Consistency condition 3 was verified.

4 Clause 4: Scaling and Data Base Population Related Items

4.1. Cardinality of Tables

The cardinality (e.g., the number of rows) of each table, as it existed at the start of the benchmark run, must be disclosed.

Table 4-1 portrays the TPC Benchmark™ C defined tables and the number of rows for each table as they were built initially.

All tables are based on 81,600 warehouses, the number of active warehouses during the benchmark.

Table Name	Number of Rows
Warehouse	81,600
District	816,000
Customer	2,448,000,000
History	2,448,000,000
Order	2,448,000,000
New Order	734,400,000
Order Line	24,479,025,798
Stock	8,160,000,000
Item	100,000

Table 4-1: Initial Cardinality of Tables

4.2. Distribution of Tables and Logs

The distribution of tables and logs across all media must be explicitly depicted for the tested and priced systems.

A pair of FC adapters connected to a storage controller was used for the log. The storage controller contained four RAID5 arrays each containing 14 disk drives. The log logical volume was striped across the four arrays (hdisks). Each of the disks used for the log had 36GB of storage capacity.

There are 10 pairs of FC adapters connected to 10 storage controllers. Eight of the storage controllers contained 163 disks and two contained 164 disks for a total of 1632 disks. All storage controllers were used as evenly as possible.

All database data was evenly distributed on 51 storage volume groups. Each volume group is created using 32 disks. Each one of the 51 volume groups corresponds with one database partition. Each group contains 12 logical volumes and each volume corresponds to one DB2 container.

RAID0 was used to create the disk arrays used for the volume groups.

4.3. Data Base Model Implemented

A statement must be provided that describes the data base model implemented by the DBMS used.

The database manager used for this testing was DB2 UDB 8.2. DB2 UDB is a relational DBMS. DB2 remote stored procedures and embedded SQL statements were used. The DB2 stored procedures were invoked via SQL CALL statements. Both the client application and stored procedures were written in embedded C code.

4.4. Partitions/Replications Mapping

The mapping of data base partitions/replications must be explicitly described.

The Warehouse, District, Customer, Order, Order-Line, New Order, History and Stock tables were horizontally partitioned into multiple tables. The specifics of the distribution of partitioned and non-partitioned tables across the physical media can be found in Table 4-2:.

DATA DISTRIBUTION		
VOLUME GROUP NAME	DATABASE PARTITION	LOGICAL VOLUMES
F01V1	1	D1F01V1ITEM, D1F01V1WAE, D1F01V1DIST, D1F01V1CSTI, D1F01V1NORA, D1F01V1ORL, D1F01V1STK, D1F01V1CST, D1F01V1ORDI, D1F01V1ORD, D1F01V1HIST, D1F01V1NORB
F01V2	2	D1F01V2ITEM, D1F01V2WARE, D1F01V2DIST, D1F01V2CSTI, D1F01V2NORA, D1F01V2ORL, D1F01V2STK, D1F01V2CST, D1F01V2ORDI, D1F01V2ORD, D1F01V2HIST, D1F01V2NORB
F01V3	3	D1F01V3ITEM, D1F01V3WARE, D1F01V3DIST, D1F01V3CSTI, D1F01V3NORA, D1F01V3ORL, D1F01V3STK, D1F01V3CST, D1F01V3ORDI, D1F01V3ORD, D1F01V3HIST, D1F01V3NORB
F01V4	4	D1F01V4ITEM, D1F01V4WARE, D1F01V4DIST, D1F01V4CSTI, D1F01V4NORA, D1F01V4ORL, D1F01V4STK, D1F01V4CST, D1F01V4ORDI, D1F01V4ORD, D1F01V4HIST, D1F01V4NORB
F01V5	5	D1F01V5ITEM, D1F01V5WARE, D1F01V5DIST, D1F01V5CSTI, D1F01V5NORA, D1F01V5ORL, D1F01V5STK, D1F01V5CST, D1F01V5ORDI, D1F01V5ORD, D1F01V5HIST, D1F01V5NORB
F02V1	6	D1F02V1ITEM, D1F02V1WAE, D1F02V1DIST, D1F02V1CSTI, D1F02V1NORA, D1F02V1ORL, D1F02V1STK, D1F02V1CST, D1F02V1ORDI, D1F02V1ORD, D1F02V1HIST, D1F02V1NORB
F02V2	7	D1F02V2ITEM, D1F02V2WARE, D1F02V2DIST, D1F02V2CSTI, D1F02V2NORA, D1F02V2ORL, D1F02V2STK, D1F02V2CST, D1F02V2ORDI, D1F02V2ORD, D1F02V2HIST, D1F02V2NORB
F02V3	8	D1F02V3ITEM, D1F02V3WARE, D1F02V3DIST, D1F02V3CSTI, D1F02V3NORA, D1F02V3ORL, D1F02V3STK, D1F02V3CST, D1F02V3ORDI, D1F02V3ORD, D1F02V3HIST, D1F02V3NORB
F02V4	9	D1F02V4ITEM, D1F02V4WARE, D1F02V4DIST, D1F02V4CSTI, D1F02V4NORA, D1F02V4ORL, D1F02V4STK, D1F02V4CST, D1F02V4ORDI, D1F02V4ORD, D1F02V4HIST, D1F02V4NORB
F02V5	10	D1F02V5ITEM, D1F02V5WARE, D1F02V5DIST, D1F02V5CSTI, D1F02V5NORA, D1F02V5ORL, D1F02V5STK, D1F02V5CST, D1F02V5ORDI, D1F02V5ORD, D1F02V5HIST, D1F02V5NORB
F03V1	11	D1F03V1ITEM, D1F03V1WAE, D1F03V1DIST, D1F03V1CSTI, D1F03V1NORA, D1F03V1ORL, D1F03V1STK, D1F03V1CST, D1F03V1ORDI, D1F03V1ORD, D1F03V1HIST, D1F03V1NORB
F03V2	12	D1F03V2ITEM, D1F03V2WARE, D1F03V2DIST, D1F03V2CSTI, D1F03V2NORA, D1F03V2ORL, D1F03V2STK, D1F03V2CST, D1F03V2ORDI, D1F03V2ORD, D1F03V2HIST, D1F03V2NORB
F03V3	13	D1F03V3ITEM, D1F03V3WARE, D1F03V3DIST, D1F03V3CSTI, D1F03V3NORA, D1F03V3ORL, D1F03V3STK, D1F03V3CST, D1F03V3ORDI, D1F03V3ORD, D1F03V3HIST, D1F03V3NORB
F03V4	14	D1F03V4ITEM, D1F03V4WARE, D1F03V4DIST, D1F03V4CSTI, D1F03V4NORA, D1F03V4ORL, D1F03V4STK, D1F03V4CST, D1F03V4ORDI, D1F03V4ORD, D1F03V4HIST, D1F03V4NORB

F03V5	15	D1F03V5ITEM, D1F03V5WARE, D1F03V5DIST, D1F03V5CSTI, D1F03V5NORA, D1F03V5ORL, D1F03V5STK, D1F03V5CST, D1F03V5ORDI, D1F03V5ORD, D1F03V5HIST, D1F03V5NORB
F04V1	16	D1F04V1ITEM, D1F04V1WAE, D1F04V1DIST, D1F04V1CSTI, D1F04V1NORA, D1F04V1ORL, D1F04V1STK, D1F04V1CST, D1F04V1ORDI, D1F04V1ORD, D1F04V1HIST, D1F04V1NORB
F04V2	17	D1F04V2ITEM, D1F04V2WARE, D1F04V2DIST, D1F04V2CSTI, D1F04V2NORA, D1F04V2ORL, D1F04V2STK, D1F04V2CST, D1F04V2ORDI, D1F04V2ORD, D1F04V2HIST, D1F04V2NORB
F04V3	18	D1F04V3ITEM, D1F04V3WARE, D1F04V3DIST, D1F04V3CSTI, D1F04V3NORA, D1F04V3ORL, D1F04V3STK, D1F04V3CST, D1F04V3ORDI, D1F04V3ORD, D1F04V3HIST, D1F04V3NORB
F04V4	19	D1F04V4ITEM, D1F04V4WARE, D1F04V4DIST, D1F04V4CSTI, D1F04V4NORA, D1F04V4ORL, D1F04V4STK, D1F04V4CST, D1F04V4ORDI, D1F04V4ORD, D1F04V4HIST, D1F04V4NORB
F04V5	20	D1F04V5ITEM, D1F04V5WARE, D1F04V5DIST, D1F04V5CSTI, D1F04V5NORA, D1F04V5ORL, D1F04V5STK, D1F04V5CST, D1F04V5ORDI, D1F04V5ORD, D1F04V5HIST, D1F04V5NORB
F05V1	21	D1F05V1ITEM, D1F05V1WAE, D1F05V1DIST, D1F05V1CSTI, D1F05V1NORA, D1F05V1ORL, D1F05V1STK, D1F05V1CST, D1F05V1ORDI, D1F05V1ORD, D1F05V1HIST, D1F05V1NORB
F05V2	22	D1F05V2ITEM, D1F05V2WARE, D1F05V2DIST, D1F05V2CSTI, D1F05V2NORA, D1F05V2ORL, D1F05V2STK, D1F05V2CST, D1F05V2ORDI, D1F05V2ORD, D1F05V2HIST, D1F05V2NORB
F05V3	23	D1F05V3ITEM, D1F05V3WARE, D1F05V3DIST, D1F05V3CSTI, D1F05V3NORA, D1F05V3ORL, D1F05V3STK, D1F05V3CST, D1F05V3ORDI, D1F05V3ORD, D1F05V3HIST, D1F05V3NORB
F05V4	24	D1F05V4ITEM, D1F05V4WARE, D1F05V4DIST, D1F05V4CSTI, D1F05V4NORA, D1F05V4ORL, D1F05V4STK, D1F05V4CST, D1F05V4ORDI, D1F05V4ORD, D1F05V4HIST, D1F05V4NORB
F05V5	25	D1F05V5ITEM, D1F05V5WARE, D1F05V5DIST, D1F05V5CSTI, D1F05V5NORA, D1F05V5ORL, D1F05V5STK, D1F05V5CST, D1F05V5ORDI, D1F05V5ORD, D1F05V5HIST, D1F05V5NORB
F06V1	26	D1F06V1ITEM, D1F06V1WAE, D1F06V1DIST, D1F06V1CSTI, D1F06V1NORA, D1F06V1ORL, D1F06V1STK, D1F06V1CST, D1F06V1ORDI, D1F06V1ORD, D1F06V1HIST, D1F06V1NORB
F06V2	27	D1F06V2ITEM, D1F06V2WARE, D1F06V2DIST, D1F06V2CSTI, D1F06V2NORA, D1F06V2ORL, D1F06V2STK, D1F06V2CST, D1F06V2ORDI, D1F06V2ORD, D1F06V2HIST, D1F06V2NORB
F06V3	28	D1F06V3ITEM, D1F06V3WARE, D1F06V3DIST, D1F06V3CSTI, D1F06V3NORA, D1F06V3ORL, D1F06V3STK, D1F06V3CST, D1F06V3ORDI, D1F06V3ORD, D1F06V3HIST, D1F06V3NORB
F06V4	29	D1F06V4ITEM, D1F06V4WARE, D1F06V4DIST, D1F06V4CSTI, D1F06V4NORA, D1F06V4ORL, D1F06V4STK, D1F06V4CST, D1F06V4ORDI, D1F06V4ORD, D1F06V4HIST, D1F06V4NORB
F06V5	30	D1F06V5ITEM, D1F06V5WARE, D1F06V5DIST, D1F06V5CSTI, D1F06V5NORA, D1F06V5ORL, D1F06V5STK, D1F06V5CST, D1F06V5ORDI, D1F06V5ORD, D1F06V5HIST, D1F06V5NORB
F07V1	31	D1F07V1ITEM, D1F07V1WAE, D1F07V1DIST, D1F07V1CSTI, D1F07V1NORA, D1F07V1ORL, D1F07V1STK, D1F07V1CST, D1F07V1ORDI, D1F07V1ORD, D1F07V1HIST, D1F07V1NORB
F07V2	32	D1F07V2ITEM, D1F07V2WARE, D1F07V2DIST, D1F07V2CSTI, D1F07V2NORA, D1F07V2ORL, D1F07V2STK, D1F07V2CST, D1F07V2ORDI, D1F07V2ORD, D1F07V2HIST, D1F07V2NORB

F07V3	33	D1F07V3ITEM, D1F07V3WARE, D1F07V3DIST, D1F07V3CSTI, D1F07V3NORA, D1F07V3ORL, D1F07V3STK, D1F07V3CST, D1F07V3ORDI, D1F07V3ORD, D1F07V3HIST, D1F07V3NORB
F07V4	34	D1F07V4ITEM, D1F07V4WARE, D1F07V4DIST, D1F07V4CSTI, D1F07V4NORA, D1F07V4ORL, D1F07V4STK, D1F07V4CST, D1F07V4ORDI, D1F07V4ORD, D1F07V4HIST, D1F07V4NORB
F07V5	35	D1F07V5ITEM, D1F07V5WARE, D1F07V5DIST, D1F07V5CSTI, D1F07V5NORA, D1F07V5ORL, D1F07V5STK, D1F07V5CST, D1F07V5ORDI, D1F07V5ORD, D1F07V5HIST, D1F07V5NORB
F08V1	36	D1F08V1ITEM, D1F08V1WAE, D1F08V1DIST, D1F08V1CSTI, D1F08V1NORA, D1F08V1ORL, D1F08V1STK, D1F08V1CST, D1F08V1ORDI, D1F08V1ORD, D1F08V1HIST, D1F08V1NORB
F08V2	37	D1F08V2ITEM, D1F08V2WARE, D1F08V2DIST, D1F08V2CSTI, D1F08V2NORA, D1F08V2ORL, D1F08V2STK, D1F08V2CST, D1F08V2ORDI, D1F08V2ORD, D1F08V2HIST, D1F08V2NORB
F08V3	38	D1F08V3ITEM, D1F08V3WARE, D1F08V3DIST, D1F08V3CSTI, D1F08V3NORA, D1F08V3ORL, D1F08V3STK, D1F08V3CST, D1F08V3ORDI, D1F08V3ORD, D1F08V3HIST, D1F08V3NORB
F08V4	39	D1F08V4ITEM, D1F08V4WARE, D1F08V4DIST, D1F08V4CSTI, D1F08V4NORA, D1F08V4ORL, D1F08V4STK, D1F08V4CST, D1F08V4ORDI, D1F08V4ORD, D1F08V4HIST, D1F08V4NORB
F08V5	40	D1F08V5ITEM, D1F08V5WARE, D1F08V5DIST, D1F08V5CSTI, D1F08V5NORA, D1F08V5ORL, D1F08V5STK, D1F08V5CST, D1F08V5ORDI, D1F08V5ORD, D1F08V5HIST, D1F08V5NORB
F09V1	41	D1F09V1ITEM, D1F09V1WAE, D1F09V1DIST, D1F09V1CSTI, D1F09V1NORA, D1F09V1ORL, D1F09V1STK, D1F09V1CST, D1F09V1ORDI, D1F09V1ORD, D1F09V1HIST, D1F09V1NORB
F09V2	42	D1F09V2ITEM, D1F09V2WARE, D1F09V2DIST, D1F09V2CSTI, D1F09V2NORA, D1F09V2ORL, D1F09V2STK, D1F09V2CST, D1F09V2ORDI, D1F09V2ORD, D1F09V2HIST, D1F09V2NORB
F09V3	43	D1F09V3ITEM, D1F09V3WARE, D1F09V3DIST, D1F09V3CSTI, D1F09V3NORA, D1F09V3ORL, D1F09V3STK, D1F09V3CST, D1F09V3ORDI, D1F09V3ORD, D1F09V3HIST, D1F09V3NORB
F09V4	44	D1F09V4ITEM, D1F09V4WARE, D1F09V4DIST, D1F09V4CSTI, D1F09V4NORA, D1F09V4ORL, D1F09V4STK, D1F09V4CST, D1F09V4ORDI, D1F09V4ORD, D1F09V4HIST, D1F09V4NORB
F09V5	45	D1F09V5ITEM, D1F09V5WARE, D1F09V5DIST, D1F09V5CSTI, D1F09V5NORA, D1F09V5ORL, D1F09V5STK, D1F09V5CST, D1F09V5ORDI, D1F09V5ORD, D1F09V5HIST, D1F09V5NORB
F10V1	46	D1F10V1ITEM, D1F10V1WAE, D1F10V1DIST, D1F10V1CSTI, D1F10V1NORA, D1F10V1ORL, D1F10V1STK, D1F10V1CST, D1F10V1ORDI, D1F10V1ORD, D1F10V1HIST, D1F10V1NORB
F10V2	47	D1F10V2ITEM, D1F10V2WARE, D1F10V2DIST, D1F10V2CSTI, D1F10V2NORA, D1F10V2ORL, D1F10V2STK, D1F10V2CST, D1F10V2ORDI, D1F10V2ORD, D1F10V2HIST, D1F10V2NORB
F10V3	48	D1F10V3ITEM, D1F10V3WARE, D1F10V3DIST, D1F10V3CSTI, D1F10V3NORA, D1F10V3ORL, D1F10V3STK, D1F10V3CST, D1F10V3ORDI, D1F10V3ORD, D1F10V3HIST, D1F10V3NORB
F10V4	49	D1F10V4ITEM, D1F10V4WARE, D1F10V4DIST, D1F10V4CSTI, D1F10V4NORA, D1F10V4ORL, D1F10V4STK, D1F10V4CST, D1F10V4ORDI, D1F10V4ORD, D1F10V4HIST, D1F10V4NORB
F10V5	50	D1F10V5ITEM, D1F10V5WARE, D1F10V5DIST, D1F10V5CSTI, D1F10V5NORA, D1F10V5ORL, D1F10V5STK, D1F10V5CST, D1F10V5ORDI, D1F10V5ORD, D1F10V5HIST, D1F10V5NORB

F11V1	51	D1F11V1ITEM, D1F11V1WAE, D1F11V1DIST, D1F11V1CSTI, D1F11V1NORA, D1F11V1ORL, D1F11V1STK, D1F11V1CST, D1F11V1ORDI, D1F11V1ORD, D1F11V1HIST, D1F11V1NORB
-------	----	---

Table 4-2: IBM System p5 570 Model 9117-570 Data Distribution Benchmark Configuration

4.5. 60-Day Space Calculations

Details of the 60 day space computations along with proof that the database is configured to sustain 8 hours of growth for the dynamic tables (Order, Order-Line, and History) must be disclosed.

Warehouses	81,600				
Measured TpmC	1,025,170				
Table	Rows	Table	Index	5% Space	Total Space
Warehouse	81,600		38	0	2
District	816,000		306	0	15
Item	100,000		10	0	1
Stock	8,160,000,000	2,656,488		0	132,824
Customer	2,448,000,000	1,912,755	117,836		101,530
New-Order	734,400,000	56,814		0	2,841
Orders	2,448,000,000	110,466	68,468		0
Order-Line	24,480,000,000	1,969,620		0	0
History	2,448,000,000	151,572		0	0
Additional Overhead		984,810			
Free Space	266,557				
Dynamic Space	2,231,658				
Static Space	6,034,737				
Daily Growth	448,594				
Daily Spread	0				
				<u>30 Minute log Computations</u>	
				Log Written (KB)	72,889,565
				New-Order Txns	30,755,091
				Log Written per New-Order (KB)	2.37
Data Storage Requirement					
60 Days (MB)	32,950,362				
60 Days (GB)	32,178				
Log Storage Requirement					
8 Hours (GB)	1,112.21				
Disk Sizing					
	Formatted	SUT	Priced		
Disk Type	Capacity (GB)	# of Disks	Capacity (GB)	# of Disks	Capacity (GB)
DB Storage	36.40	1,632	59,405	1,632	59,405
LOG Storage	312.50	4	1,250	4	1,250
OS Storage	72.40	6	434	6	434
Total Capacity					61,089

5 Clause 5: Performance Metrics and Response Time Related Items

5.1. Response Times

Ninetieth percentile, maximum and average response times must be reported for all transaction types as well as for the Menu response time.

Table 5-1 lists the response times and the ninetieth percentiles for each of the transaction types for the measured system.

5.2. Keying and Think Times

The minimum, the average, and the maximum keying and think times must be reported for each transaction type.

Table 5-1 lists the TPC-C keying and think times for the measured system.

Response Times	New Order	Payment	Order Status	Delivery (int./def.)	Stock Level	Menus
90 %	0.50	0.50	0.50	0.25/0.80	0.65	0.23
Average	0.34	0.34	0.34	0.16/0.15	0.35	0.16
Maximum	6.84	7.17	6.90	6.50/6/16	6.16	6.51
Think Times						
Minimum	0.01	0.01	0.01	0.01	0.01	N/A
Average	12.02	12.02	10.01	5.02	5.02	N/A
Maximum	120.20	120.21	100.10	50.20	50.20	N/A
Keying Times						
Minimum	18.00	3.00	2.00	2.00	2.00	N/A
Average	18.01	3.01	2.01	2.01	2.01	N/A
Maximum	18.05	3.05	2.04	2.05	2.05	N/A

Table 5-1: Think and Keying Times

5.3. Response Time Frequency Distribution

Response time frequency distribution curves must be reported for each transaction type.

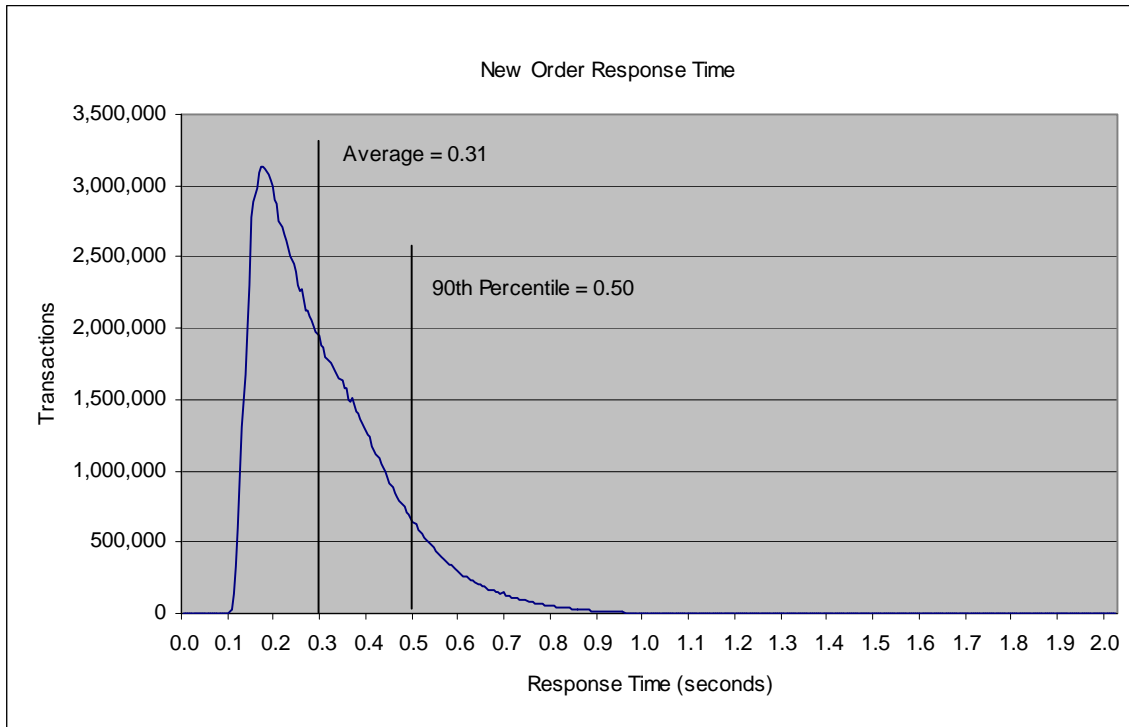


Figure 5-1: New-Order Response Time Distribution

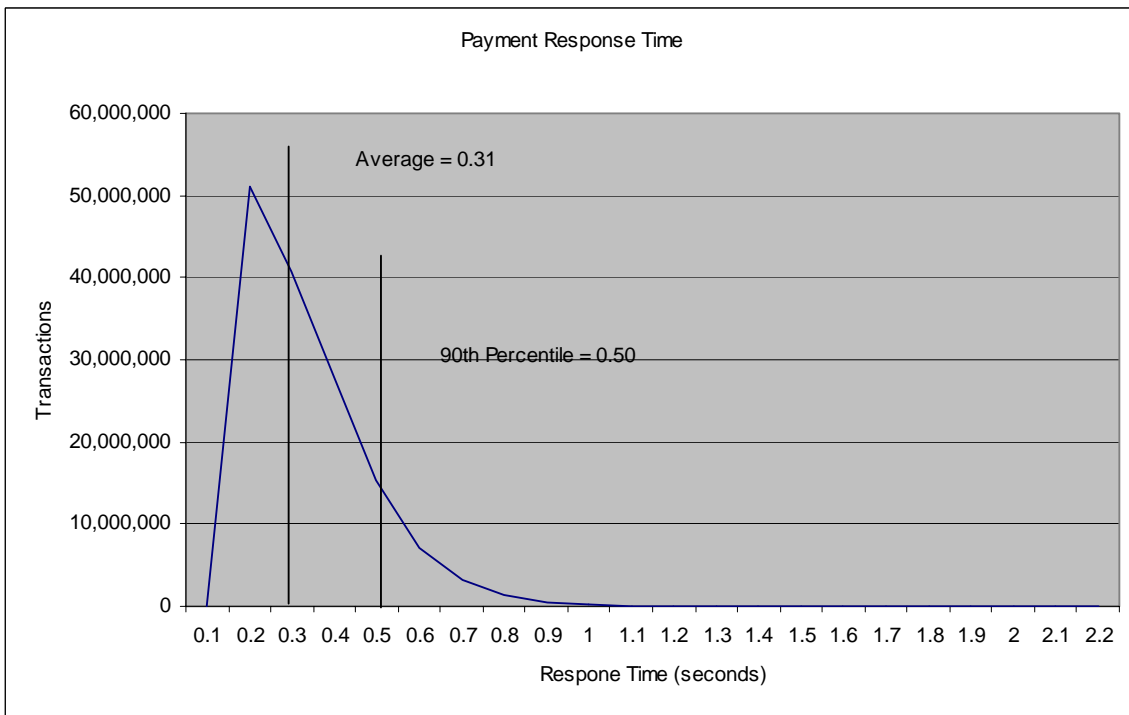


Figure 5-2: Payment Response Time Distribution

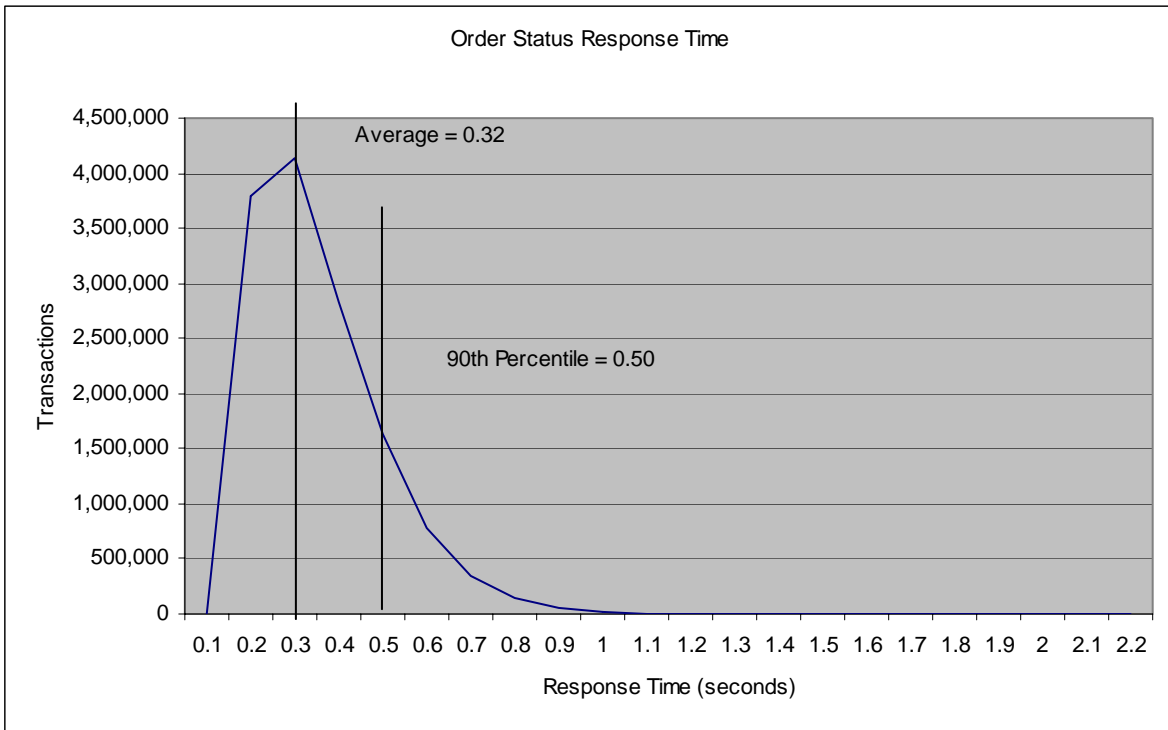


Figure 5-3: Order-Status Response Time Distribution

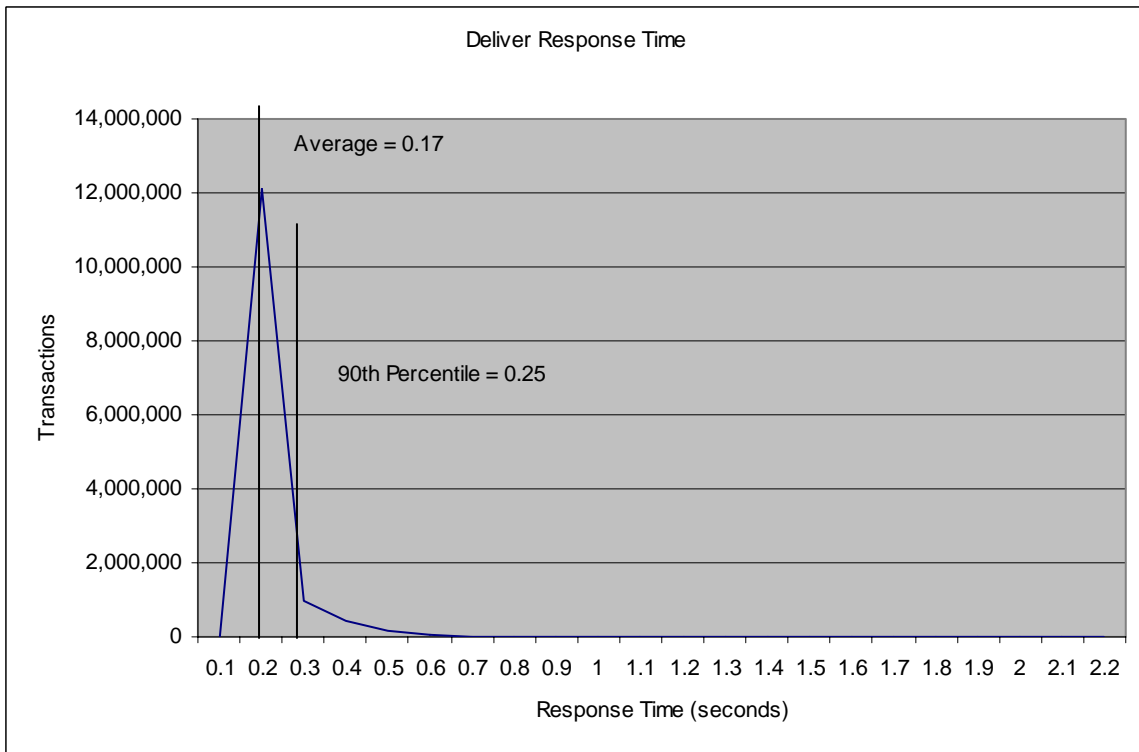


Figure 5-4: Delivery (Interactive) Response Time Distribution

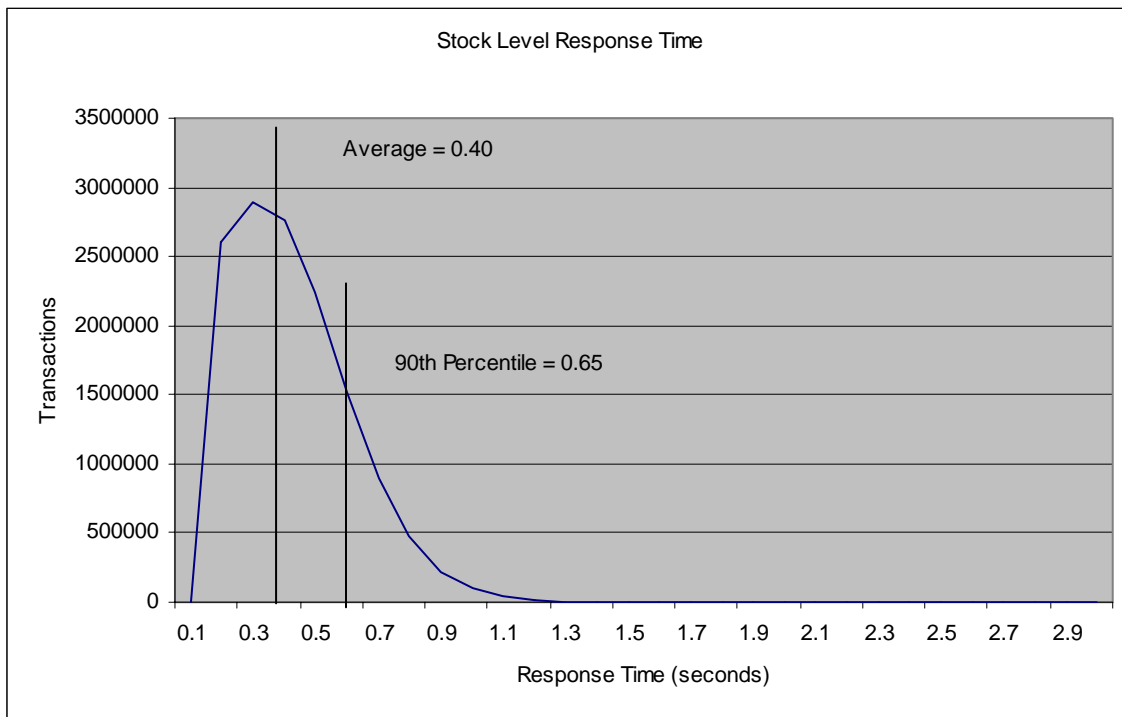


Figure 5-5: Stock Level Response Time Distribution

5.4. Performance Curve for Response Time versus Throughput

The performance curve for response times versus throughput must be reported for the New-Order transaction.

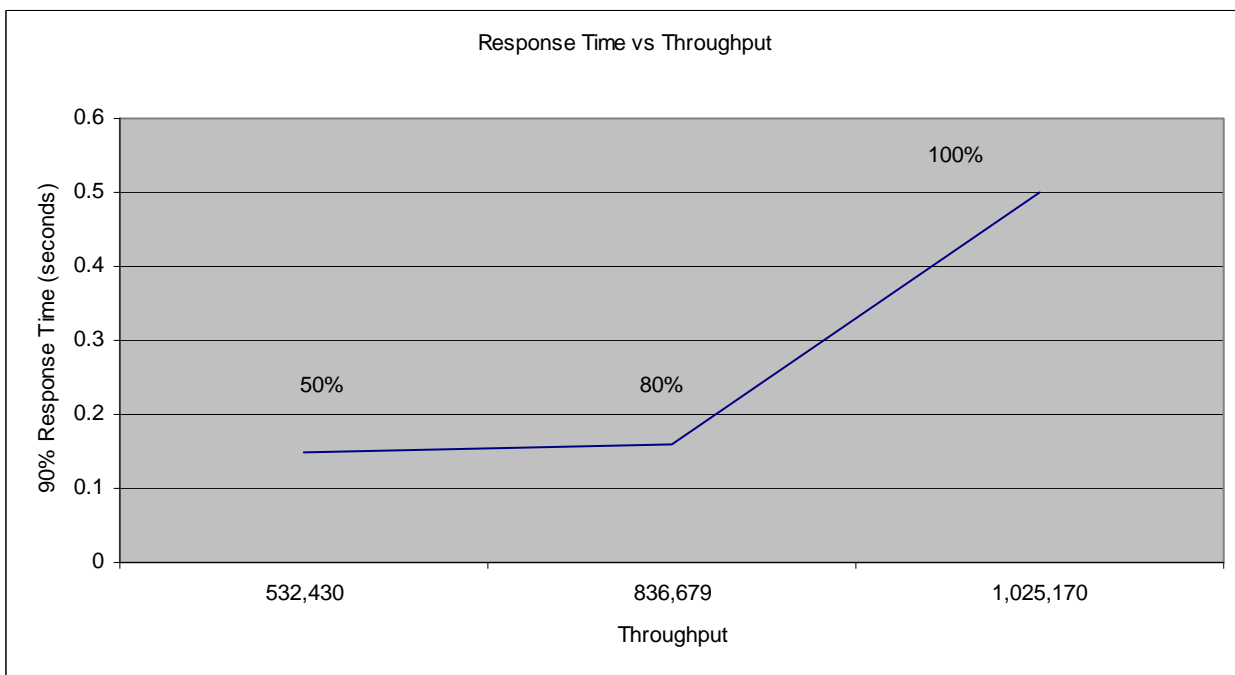


Figure 5-6: New-Order Response Time vs. Throughput

5.5. Think Time Frequency Distribution

A graph of the think time frequency distribution must be reported for the New-Order transaction.

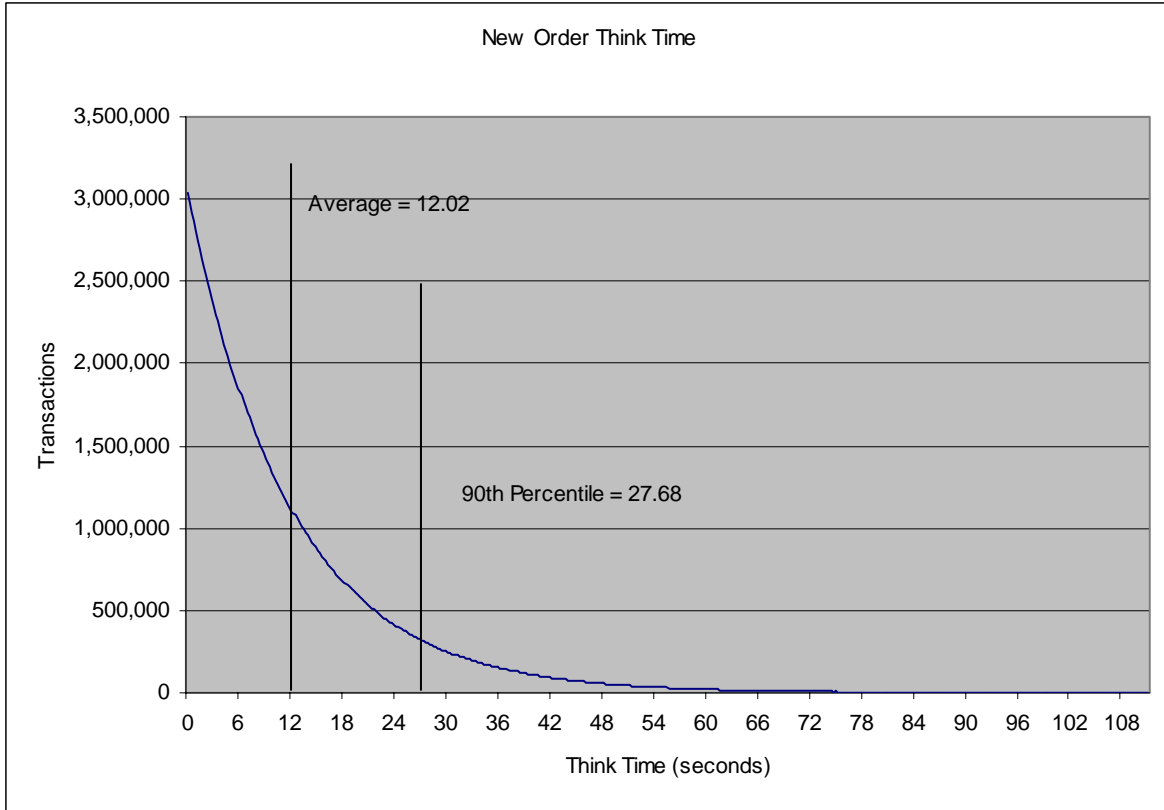


Figure 5-7: New-Order Think Time Distribution

5.6. Throughput versus Elapsed Time

A graph of throughput versus elapsed time must be reported for the New-Order transaction.

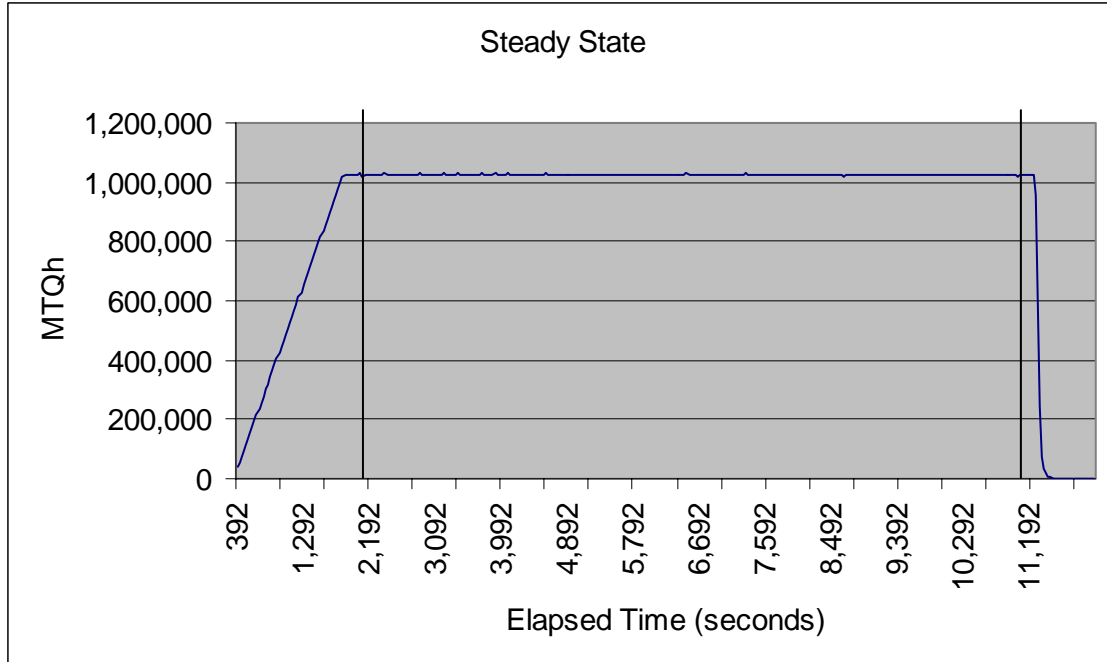


Figure 5-8: New-Order Throughput vs. Elapsed Time

5.7. Steady State Determination

The method used to determine that the SUT had reached a steady state prior to commencing the measurement interval must be described.

All the emulated users were allowed to logon and do transactions. The user ramp-up phase is clearly visible on the graph above. Refer to the Numerical Quantities Summary pages for the rampup time. Figure 5-8 New-Order throughput versus Elapsed Time graph shows that the system maintained a steady state during the measurement interval

5.8. Work Performed During Steady State

A description of how the work normally performed during a sustained test (for example check pointing, writing redo/undo log records, etc), actually occurred during the measurement interval must be reported.

A 2-hour 30-minute measurement interval was used to guaranty that all work normally performed during an 8-hour sustained test are included in the reported throughput.

5.8.1. Transaction Flow

Each of the 4 (non-delivery) transactions is serviced by 2 individual programs, Internet Information System 5.1 (IIS) and a Microsoft COM+ 1.0 Queued Component Server, used as the transaction manager (COM+). Both programs are running on the client system:

- The initial HTML 1.0 request is serviced by an ISAPI custom-written handler running on Internet Information System 5.1(IIS). IIS is responsible for handling all HTML requests. The web server communicates to the COM+ server through a Microsoft COM+ api interface.
- COM+ communicates with the Server system over Ethernet and handles all database operations, using DB2 embedded SQL calls.

When the COM+ server boots up, it creates a configurable amount of connections to the Server (listed in application settings).

COM+ routes the transaction and balances the load according to the options defined in the Component Services GUI for the COM+ server application and settings in the Windows 2000 Registry. The configuration file and registry variables are listed in Appendix B.2.

At the beginning, each TPC-C user sends a pair of HTML 1.0 requests submitting the its unique warehouse and district to the IIS ISAPI handler. Upon successful validation of user's login, IIS the displays an HTML form which encapsulates the TPC-C transaction menu.

The transaction flow is described below:

- The TPC-C user requests the transaction type's HTML form and proceeds to generate (fill in) a GET request with the required files for the transaction.
- IIS accepts the filled in GET request , parses, and validates all values entered by the user.
- It then proceeds to transmit those values to the COM+ server through an transaction type specific COM+ api interface.
- The COM+ Pool Manager receives the request and first decides if there is a connection object in the pool available to service it.
 - If so, the connection is used to send the transaction request to the Server.
 - If no connection is available, the request will enter a COM+ internal queue and will be serviced by the next available connection.
- Once the connection is available to be used, a COM+ pool thread receives the transaction and calls a TPC-C back end DB2 client api to execute all database operations related to the transaction type. (All the transaction information entered on the HTML form is available in a data structure provided by the ISAPI caller).
- The transaction is committed and the DB2 back end client returns control back to the COM pool thread.
- COM pool thread returns control to the ISAPI caller.
(All transaction results are inside the data structure that the ISAPI caller provided to the COM+ api in the parameter list).
- ISAPI caller returns control to the "screen application" by doing a PUT request.

5.8.2. Database Transaction

All database operations are performed by the TPC-C back-end programs. The process is described below:

Using embedded SQL calls, the TPC-C back-end program interacts with DB2 UDB Server to perform SQL data manipulations such as update, select, delete and insert, as required by the transaction. After all database operations are performed for a transaction, the transaction is committed.

DB2 UDB Server proceeds to update the database as follows:

When DB2 UDB Server changes a database table with an update, insert, or delete operation, the change is initially made in memory, not on disk. When there is not enough space in the memory buffer to read in or write additional data pages, DB2 UDB Server will make space by flushing some modified pages to disk. Modified pages are also written to disk as part of the "Soft" checkpoint to ensure that no updates remain unflushed for longer than the allowed time. Before a change is made to the database, it is first recorded in the transaction log. This ensures that the database can be recovered completely in the event of a failure. Using the transaction log, transactions that started but did not complete prior to a failure can be undone, and transactions recorded as complete in the transaction log but not yet written to disk can be redone.

5.8.3. Checkpoints

DB2 UDB uses a write-ahead-logging protocol to guarantee recovery. This protocol uses "Soft" checkpoint to write least-recently-used database pages to disk independent of transaction commit. However, enough log information to redo/undo the change to a database pages is committed to disk before the database page itself is written. This protocol therefore renders checkpoint unnecessary for DB2 UDB. For a more detailed description of the general principles of the write-ahead-logging protocol, see the IBM research paper, "ARIES: A Transaction Recovery Method Supporting Fine Granularity Locking and Partial Rollbacks Using Write-Ahead Logging," by C. Mohan, Database Technology Institute, IBM Almaden Research Center.

([http:// portal.acm.org/citation.cfm?id=128770&coll=portal&dl=ACM&CFID=10343790&CFTOKEN=42047146](http://portal.acm.org/citation.cfm?id=128770&coll=portal&dl=ACM&CFID=10343790&CFTOKEN=42047146))

5.9. Measurement Interval

A statement of the duration of the measurement interval for the reported Maximum Qualified Throughput (tpmC) must be included.

A 2-hour 30-minute measurement interval was used. No connections were lost during the run.

6 Clause 6: SUT, Driver, and Communication Definition Related Items

6.1. RTE Availability

If the RTE is commercially available, then its inputs must be specified. Otherwise, a description must be supplied of what inputs to the RTE had been used.

IBM used an internally developed RTE for these tests. Appendix D contains the scripts used in the testing.

6.2. Functionality and Performance of Emulated Components

It must be demonstrated that the functionality and performance of the components being emulated in the Driver System are equivalent to that of the priced system.

No components were emulated.

6.3. Network Bandwidth

The bandwidth of the network(s) used in the tested/priced configuration must be disclosed.

The database system was connected to 4 Ethernet Gigabit switches each with a rate of 1000Mbits full-duplex. Each group of 10 clients were connected to one of the Gigabit Ethernet switches at 1000Mbits full-duplex rate.

6.4. Operator Intervention

If the configuration requires operator intervention, the mechanism and the frequency of this intervention must be disclosed.

No operator intervention is required to sustain the reported throughput during the eight-hour period.

7 Clause 7: Pricing Related Items

7.1. Hardware and Programs Used

A detailed list of the hardware and software used in the priced system must be reported. Each item must have vendor part number, description, and release/revision level, and either general availability status or committed delivery date. If package-pricing is used, contents of the package must be disclosed. Pricing source(s) and effective date(s) must also be reported.

The detailed list of all hardware and programs for the priced configuration is listed in the pricing sheets (please refer to Section 8.2 for details) for each system reported. The prices for all products and features that are provided by IBM are available the same day as product or feature availability.

7.2. Three Year Cost of System Configuration

The total 3-year price of the entire configuration must be reported, including: hardware, software, and maintenance charges. Separate component pricing is recommended. The basis of all discounts used must be disclosed.

The pricing details for this disclosure is contained in the executive summary pages. All 3rd party quotations are included at the end of this report in Appendix E. All prices are based on IBM US list prices.

Discount are based on US list prices and for similar quantities and configurations.

7.3. Availability Dates

The committed delivery date for general availability (availability date) of products used in the price calculations must be reported. When the priced system includes products with different availability dates, the reported availability date for the priced system must be the date at which all components are committed to be available.

All components of the SUT will be available on: May 31, 2006.

7.4. Statement of tpmC and Price/Performance

A statement of the measured tpmC, as well as the respective calculations for 3-year pricing, price/performance (price/tpmC), and the availability date must be disclosed.

.System	tpmC	3-year System Cost	\$/tpmC	Availability Date
IBM System p5 570 Model 9117-570	1,025,169.69	\$4,528,937 USD	\$4.42 USD	May 31, 2006

Please refer to the price list on the Executive Summary page for details.

8 Clause 9: Audit Related Items

If the benchmark has been independently audited, then the auditor's name, address, phone number, and a brief audit summary report indicating compliance must be included in the Full Disclosure Report. A statement should be included, specifying when the complete audit report will become available and who to contact in order to obtain a copy.

The auditor's attestation letter is included in this section of this report:

Benchmark Sponsor: John J. Makis
 IBM eServer Performance
 11501 Burnet Road
 Austin, TX 78758

January 27, 2006

I verified the TPC Benchmark™ C performance of the following Client Server configuration:

Platform: IBM eServer p5 570 Model 9117-570 c/s
 Operating system: AIX 5L Version 5.3
 Database Manager: DB2 Universal Database 8.2
 Transaction Manager: Microsoft COM+

The results were:

CPU's (Speed)	Memory	Disks	NewOrder 90% Response Time	tpmC
Server: IBM eServer p5 570 570 Model 9117-570				
16 x POWRR5+ (2.2GHz)	512 GB (8 x 36 MB L3)	1688 x 36.4 GB FC 6 x 72.4 GB SCSI	0.50 Seconds	1,025,169.69
Fourty Clients: IBM eServer xSeries 336 (each with)				
2 x Xeon (3.2 GHz)	2.0 GB (2 MB cache)	1 x 36 GB u320 SCSI	n/a	n/a

In my opinion, these performance results were produced in compliance with the TPC requirements for the benchmark.

The following verification items were given special attention:

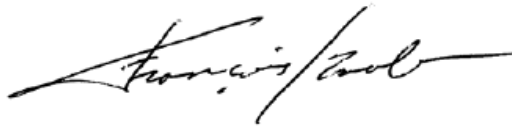
- The transactions were correctly implemented
- The database records were the proper size
- The database was properly scaled and populated

- The ACID properties were met
- Input data was generated according to the specified percentages
- The transaction cycle times included the required keying and think times
- The reported response times were correctly measured.
- At least 90% of all delivery transactions met the 80 Second completion time limit
- All 90% response times were under the specified maximums
- The measurement interval was representative of steady state conditions
- The reported measurement interval was 120 minutes
- Write-ahead-logging was active during the measurement interval
- The 60 day storage requirement was correctly computed
- The system pricing was verified for major components and maintenance

Additional Audit Notes:

None.

Respectfully Yours,

A handwritten signature in black ink, appearing to read "François Raab", with a long horizontal flourish extending to the right.

François Raab, President

Appendix - A: Client Server Code

A.1 Client/Terminal Handler Code

Makefile.config

```
#####
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp.
## 1996 - 2006
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication
## or
## disclosure restricted by GSA ADP Schedule Contract with
## IBM Corp.
#####
#####
#
# Makefile.config - NT/Win2000 Makefile Configuration
#

# Make Configuration (MSVC)
MAKE=nmake.exe

# Compiler Configuration (MSVC).
CC=cl.exe
CFLAGS_OS=-DSQLWINT -MT -DWIN32 -J -Zp8 -
DREG_KIT_METHOD -DSWAP_ENDIAN
CFLAGS_OUT=/Fo
CFLAGS_DEBUG=

# Linker Configuration (MSVC)
LD_EXEC=link.exe
LD_STORP=link.exe
LDFLAGS_EXEC=
LDFLAGS_SHLIB=/DLL
LDFLAGS_STORP=$(LDFLAGS_SHLIB) /DEF:rpctpc.def
LDFLAGS_LIB=/LIBPATH:$(TPCC_SQLLIB)\lib
/LIBPATH:"C:\Program Files\Microsoft Visual Studio\VC98\Lib"
db2api.lib winmm.lib
LDFLAGS_OUT=/OUT:

# Library Configuration
AR=lib.exe
ARFLAGS=
ARFLAGS_LIB=
ARFLAGS_OUT=/OUT:
```

```
# OS Commands
ERASE=del /F
ERASEDIR=rmdir /S
MOVE=MOVE
COPY=COPY
```

```
# OS File Extensions & Path Separator
OBJEXT=.obj
LIBEXT=.lib
SHLIBEXT=.dll
BINEXT=.exe
SLASH=\
CMDSEP=&
```

Src.Cli/Makefile

```
#####
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp.
## 1996 - 2006
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication
## or
## disclosure restricted by GSA ADP Schedule Contract with
## IBM Corp.
#####
#####
#
# Makefile - Makefile for Src.Cli (RTE/Driver Interface)
#

!include $(TPCC_ROOT)/Makefile.config

#
#####
#####
# Preprocessor, Compiler and Linker Flags
#
#####
#####

BND_OPTS = GRANT PUBLIC \
            MESSAGES $.bnd.msg
PRP_OPTS = BINDFILE \
            ISOLATION RR \
            EXPLAIN ALL \
            MESSAGES $.prep.msg \
            LEVEL $(TPCC_VERSION) \
            NOLINEMACRO

INCLUDES = -I$(TPCC_SQLLIB)/include -
I$(TPCC_ROOT)/include
```

```
CFLAGS = $(CFLAGS_OS) $(INCLUDES)
$(CFLAGS_DEBUG) \
$(UOPTS) -D$(DB2EDITION) -D$(DB2VERSION) -
D$(TPCC_SPTYPE)

OBJS =
$(TPCC_ROOT)/Src.Common/tpccmisc$(OBJEXT) \
$(TPCC_ROOT)/Src.Common/tpccdbg$(OBJEXT) \
$(TPCC_ROOT)/Src.Common/tpccctx$(OBJEXT) \
tpcccli$(OBJEXT)

LIBS = tpcccli$(LIBEXT)

#
#####
# User Targets
#
#####
all: connect $(OBJS) plan $(LIBS) disconnect
$(AR) $(ARFLAGS) $(ARFLAGS_OUT)tpcccli$(LIBEXT)
$(OBJS) $(ARFLAGS_LIB)
@echo "-----"
@echo "Please copy lval.h, db2tpcc.h, and
tpcccli$(LIBEXT) to"
@echo "a place where they can be #included and linked
with the"
@echo "RTE/driver code."
@echo "-----"

clean:
- $(ERASE) *.msg *.bnd *.plan *$(OBJEXT) *$(LIBEXT)
tpcccli.c

#
#####
# Helper Targets
#
#####

connect:
- db2 connect to $(TPCC_DBNAME)

disconnect:
- db2 connect reset
- db2 terminate

plan:
- db2exfmt -d $(TPCC_DBNAME) -e $(TPCC_SCHEMA) -
s $(TPCC_SCHEMA) -w -1 -n TPCCCLI -g -# 0 -o
TPCCCLI.exfmt.plan
- db2expln -d $(TPCC_DBNAME) -c $(TPCC_SCHEMA) -
p TPCCCLI -s 0 -g -o TPCCCLI.expln.plan

rebind: connect
db2 bind tpcccli.bnd $(BND_OPTS) QUERYOPT 7
```

```

#
#####
#####
# Build Rules
#
#####
#####

.SUFFIXES:
.SUFFIXES: $(OBJEXT) .c .sqc

tpcccli.c:
    @echo "Prepping $*.sqc"
    -db2 prep $*.sqc $(PRP_OPTS) ISOLATION RR
    @echo "Binding $*.bnd"
    db2 bind $*.bnd $(BND_OPTS) QUERYOPT 7

#
#####
#####
# Dependencies
#
#####
#####

# Client Library:
tpcccli$(LIBEXT):    $(OBJJS)

# Source
tpcc_all_sql$(OBJEXT):    tpcc_all_sql.c

# Headers
tpcc_all_sql.c:    $(TPCC_ROOT)/include/db2tpcc.h
$(TPCC_ROOT)/include/lval.h

Src.Cli/tpcccli.sqc
/*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp.
1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
*****/

/*
* tpcccli.sqc - Client/Server code for TPCC
*/

#include <stdlib.h>
#include <errno.h>
#include "db2tpcc.h"

```

```

#include "tpccapp.h"
#include "tpccdbg.h"

#include "sqlca.h"
#include "sql.h"

// -----
// New Order CLIENT
// -----

static int itemComparison ( const void * a , const void * b )
{
    struct in_items_struct * one = (struct in_items_struct *) a ;
    struct in_items_struct * two = (struct in_items_struct *) b ;

    // Primary comparison key:  I_ID
    // Secondary comparison key: W_ID

    if ( one->s_OL_I_ID != two->s_OL_I_ID )
    {
        return ( one->s_OL_I_ID - two->s_OL_I_ID ) ;
    }
    else
    {
        return ( one->s_OL_SUPPLY_W_ID - two-
>s_OL_SUPPLY_W_ID ) ;
    }
}

int neword_sql ( struct in_neword_struct * in_neword
                , struct out_neword_struct * neword )
{
    struct sqlca sqlca ;

    EXEC SQL BEGIN DECLARE SECTION;

    struct vc_new_in
    {
        short len;
        char data[ 270 ] ;
    } * pHostvarInput ;

    struct vc_new_out
    {
        short len;
        char data[ 662 ] ;
    } * pHostvarOutput ;

    EXEC SQL END DECLARE SECTION;

    int clientRc = TRAN_OK ;

    int itemIndex = 0 ;

    /* Create Timestamp */
    in_neword->s_O_ENTRY_D_time = time(NULL) ;

    // Determine if order is "all-local" or not
    // NOTE: This loop will exit on the iteration *after* finding the
last
    // item; this effectively takes care of the 0-based/1-based
conversion

```

```

// and we don't have to add one when assigning to
s_O_OL_CNT below.
    in_neword->s_all_local = 1 ;
    for ( itemIndex = 0 ;
        itemIndex < 15 && in_neword-
>in_item[ itemIndex ].s_OL_I_ID != UNUSED_ITEM_ID ;
        itemIndex++
    )
    {
        if ( in_neword-
>in_item[ itemIndex ].s_OL_SUPPLY_W_ID != in_neword-
>s_W_ID )
        {
            in_neword->s_all_local = 0 ;
        }
    }

    in_neword->s_O_OL_CNT = itemIndex ;

    // Sort the item list.  Since invalid item IDs = 100001, we will
remain
    // compliant with the spec (Section 2.4.2.3 Comment 1.

    qsort( in_neword->in_item, in_neword->s_O_OL_CNT
        , sizeof ( in_neword->in_item[ 0 ] )
        , itemComparison
        ) ;

    pHostvarInput = (struct vc_new_in *) in_neword ;
    pHostvarInput->len = sizeof(struct in_neword_struct) -
SPGENERAL_ADJUST ;

    pHostvarOutput = (struct vc_new_out *) neword;
    pHostvarOutput->len = sizeof(struct out_neword_struct) -
SPGENERAL_ADJUST ;

#ifdef DEBUGIT
    new_debug(neword, in_neword, "Client before SP call");
#endif /* DEBUGIT */

#ifdef SWAP_ENDIAN
    for (itemIndex=0; itemIndex<in_neword->s_O_OL_CNT;
itemIndex++)
    {
        SWAP_BYTE(in_neword->in_item[ itemIndex ].s_OL_I_ID);
        SWAP_BYTE(in_neword-
>in_item[ itemIndex ].s_OL_SUPPLY_W_ID);
        SWAP_BYTE(in_neword-
>in_item[ itemIndex ].s_OL_QUANTITY);
    }
    SWAP_BYTE(in_neword->s_O_ENTRY_D_time);
    SWAP_BYTE(in_neword->s_C_ID);
    SWAP_BYTE(in_neword->s_W_ID);
    SWAP_BYTE(in_neword->s_D_ID);
    SWAP_BYTE(in_neword->s_O_OL_CNT);
    SWAP_BYTE(in_neword->s_all_local);
    SWAP_BYTE(in_neword->duplicate_items);
#endif //SWAP_ENDIAN

    EXEC SQL CALL news ( :*pHostvarInput, :*pHostvarOutput ) ;

#ifdef SWAP_ENDIAN

```

```

SWAP_BYTE(in_neword->s_O_ENTRY_D_time);
SWAP_BYTE(in_neword->s_C_ID);
SWAP_BYTE(in_neword->s_W_ID);
SWAP_BYTE(in_neword->s_D_ID);
SWAP_BYTE(in_neword->s_O_OL_CNT);
SWAP_BYTE(in_neword->s_all_local);
SWAP_BYTE(in_neword->duplicate_items);
for (itemIndex=0; itemIndex<in_neword->s_O_OL_CNT;
itemIndex++)
{
    SWAP_BYTE(in_neword->in_item[ itemIndex ].s_OL_ID);
    SWAP_BYTE(in_neword-
>in_item[ itemIndex ].s_OL_SUPPLY_W_ID);
    SWAP_BYTE(in_neword-
>in_item[ itemIndex ].s_OL_QUANTITY);
}

SWAP_BYTE(neword->s_O_ENTRY_D_time);
SWAP_BYTE(neword->s_W_TAX);
SWAP_BYTE(neword->s_D_TAX);
SWAP_BYTE(neword->s_C_DISCOUNT);
SWAP_BYTE(neword->s_total_amount);
SWAP_BYTE(neword->s_O_ID);
SWAP_BYTE(neword->s_O_OL_CNT);
SWAP_BYTE(neword->s_transtatus);
SWAP_BYTE(neword->deadlocks);
for (itemIndex=0; itemIndex<in_neword->s_O_OL_CNT;
itemIndex++)
{
    SWAP_BYTE(neword->item[ itemIndex ].s_I_PRICE);
    SWAP_BYTE(neword->item[ itemIndex ].s_OL_AMOUNT);
    SWAP_BYTE(neword->item[ itemIndex ].s_S_QUANTITY);
}
#endif //SWAP_ENDIAN

if ( sqlca.sqlcode == 0 )
{
    double wtax = neword->s_W_TAX / 10000.0 ;
    double dtax = neword->s_D_TAX / 10000.0 ;
    double cdisc = neword->s_C_DISCOUNT / 10000.0 ;
    double factor = (1.0 - cdisc) * (1.0 + wtax + dtax) ;

    // Compute order total

    neword->s_total_amount = 0 ;

    for ( itemIndex = 0 ;
        itemIndex < in_neword->s_O_OL_CNT ; // from input ,
not output
        itemIndex++
        )
    {
        if ( neword->item[ itemIndex ].s_I_PRICE > 0 ) // A zero
price signifies a bad item
        {
            neword->item[ itemIndex ].s_OL_AMOUNT = neword-
>item[ itemIndex ].s_I_PRICE *
                in_neword-
>in_item[ itemIndex ].s_OL_QUANTITY ; // reference input
value

```

```

        neword->s_total_amount += neword-
>item[ itemIndex ].s_OL_AMOUNT ;
    }
}

// s_total_amount gets cast implicitly to a double to do the
arithmetic,
// and then cast back to a sqlint32.
neword->s_total_amount *= factor;
}
else
{
    sqlerror( NEWORD_SQL, "NEW", __FILE__, __LINE__,
&sqlca );
    neword->s_transtatus = FATAL_SQLERROR ;
    clientRc = FATAL_SQLERROR ;
}

/* Update Output Structure with Timestamp */
neword->s_O_ENTRY_D_time = in_neword-
>s_O_ENTRY_D_time ;

#ifdef DEBUGIT
    new_debug(neword, in_neword, "Client after SP call");
#endif /* DEBUGIT */

if (neword->s_transtatus <= FATAL_SQLERROR)
{
    new_debug(neword, in_neword, "NEW failed");
    clientRc = FATAL_SQLERROR ;
}

if (neword->s_transtatus == INVALID_ITEM)
{
    clientRc = INVALID_ITEM ;
}

return ( clientRc ) ;
}

// -----
// Payment CLIENT
// -----

int payment_sql ( struct in_payment_struct * in_payment
, struct out_payment_struct * payment )
{
    struct sqlca sqlca ;

    int clientRc = TRAN_OK ;

    EXEC SQL BEGIN DECLARE SECTION;

    // Inputs

    sqlint64 h_amount ;
    sqlint32 in_c_id ;

    struct s_data_type { short len ; char data[ 16 ] ; }
c_last_input ;

```

```

sqlint32 w_id ;
sqlint32 c_w_id ;
short d_id ;
short c_d_id ;
sqlint64 h_date ;

// Outputs

sqlint32 c_id ;

sqlint64 c_credit_lim ;
sqlint32 c_discount ;
sqlint64 c_balance ;

char w_street_1 [ 20 ] , w_street_2 [ 20 ] ;
char w_city [ 20 ] , w_state [ 2 ] , w_zip [ 9 ] ;

char d_street_1 [ 20 ] , d_street_2 [ 20 ] , d_city [ 20 ] ;
char d_state [ 2 ] , d_zip [ 9 ] , c_first [ 16 ] ;

char c_last [ 16 ] ;

char c_middle [ 2 ] , c_street_1 [ 20 ] ;
char c_street_2 [ 20 ] , c_city [ 20 ] , c_state [ 2 ] ;
char c_zip [ 9 ] , c_phone [ 16 ] ;

char c_credit [ 2 ] ;

sqlint64 c_since ;

char c_data [ 200 ] ;
short c_data_indicator = 0 ;

    struct c_data_prefix_c_last_type { short len ; char
data[ 28 ] ; } c_data_prefix_c_last ;
    struct c_data_prefix_c_id_type { short len ; char
data[ 34 ] ; } c_data_prefix_c_id ;

EXEC SQL END DECLARE SECTION;

// Input redirects

#define h_amount      in_payment->s_H_AMOUNT
#define in_c_id       in_payment->s_C_ID

#define w_id          in_payment->s_W_ID
#define d_id          in_payment->s_D_ID

#define c_d_id        in_payment->s_C_D_ID
#define c_w_id        in_payment->s_C_W_ID
#define h_date        in_payment->s_H_DATE_time

// Output redirects

#define c_credit_lim  payment->s_C_CREDIT_LIM
#define c_discount    payment->s_C_DISCOUNT
#define c_balance     payment->s_C_BALANCE

#define c_id          payment->s_C_ID
#define c_last        payment->s_C_LAST

```



```

#define c_first      payment->s_C_FIRST
#define c_middle     payment->s_C_MIDDLE
#define c_street_1  payment->s_C_STREET_1
#define c_street_2  payment->s_C_STREET_2
#define c_city       payment->s_C_CITY
#define c_state      payment->s_C_STATE
#define c_zip        payment->s_C_ZIP
#define c_phone      payment->s_C_PHONE
#define c_credit     payment->s_C_CREDIT
#define c_since      payment->s_C_SINCE_time
#define c_data       payment->s_C_DATA

#define w_street_1   payment->s_W_STREET_1
#define w_street_2   payment->s_W_STREET_2
#define w_city       payment->s_W_CITY
#define w_state      payment->s_W_STATE
#define w_zip        payment->s_W_ZIP

#define d_street_1   payment->s_D_STREET_1
#define d_street_2   payment->s_D_STREET_2
#define d_city       payment->s_D_CITY
#define d_state      payment->s_D_STATE
#define d_zip        payment->s_D_ZIP

/* Create Timestamp */
in_payment->s_H_DATE_time = (sqlint64) time( NULL );

payment->deadlocks = -1 ;
payment->s_transtatus = TRAN_OK ;

if (c_w_id == 0) { c_w_id = w_id; }
if (c_d_id == 0) { c_d_id = d_id; }

#ifdef DEBUGIT
pay_debug(payment, in_payment, "Client before SQL call");
#endif /* DEBUGIT */

// Create c_data_prefix strings and copy some elements from
// in -> out struct outside of retry_tran loop

if ( in_c_id == 0 )
{
    c_data_prefix_c_last.len =
sprintf( c_data_prefix_c_last.data, "%2.2d %6.6d %2.2d %6.6d
%04.4d.%02.2d", c_d_id , c_w_id , d_id , w_id , (int)(h_amount
/ 100) , (int)(h_amount % 100) );

    // Setup the input c_last varchar
    c_last_input.len = strlen( in_payment->s_C_LAST );
    memcpy( c_last_input.data , in_payment->s_C_LAST ,
c_last_input.len );

    // Copy to the output structure
    memcpy( payment->s_C_LAST , in_payment->s_C_LAST ,
sizeof( payment->s_C_LAST ) );

} else {

// Copy c_id to the output structure
c_id = in_c_id ;

```

```

    c_data_prefix_c_id.len = sprintf( c_data_prefix_c_id.data, "
%5.5d %2.2d %6.6d %2.2d %6.6d %04.4d.%02.2d", c_id ,
c_d_id , c_w_id , d_id , w_id , (int)(h_amount / 100),
(int)(h_amount % 100) );

}

retry_tran:

payment->deadlocks ++ ;

if ( in_c_id == 0 )
{
    EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC

        SELECT W_STREET_1, W_STREET_2, W_CITY,
W_STATE, W_ZIP
        , D_STREET_1, D_STREET_2, D_CITY,
D_STATE, D_ZIP
        , C_ID, C_FIRST, C_MIDDLE, C_STREET_1,
C_STREET_2
        , C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE,
C_CREDIT, C_CREDIT_LIM
        , C_DISCOUNT, C_BALANCE, C_DATA

    INTO :w_street_1 , :w_street_2 , :w_city , :w_state , :w_zip
        , :d_street_1 , :d_street_2 , :d_city , :d_state , :d_zip
        , :c_id , :c_first , :c_middle , :c_street_1 , :c_street_
2 , :c_city , :c_state
        , :c_zip , :c_phone , :c_since , :c_credit , :c_credit_li
m
        , :c_discount , :c_balance , :c_data :c_data_indicator

    FROM TABLE ( PAY_C_LAST( :w_id
        , :d_id
        , :c_w_id
        , :c_d_id
        , :c_last_input
        , :h_date
        , :h_amount
        , :c_data_prefix_c_last
        )
        ) AS T ( W_STREET_1, W_STREET_2,
W_CITY, W_STATE, W_ZIP
        , D_STREET_1, D_STREET_2, D_CITY,
D_STATE, D_ZIP
        , C_ID, C_FIRST, C_MIDDLE,
C_STREET_1, C_STREET_2
        , C_CITY, C_STATE, C_ZIP, C_PHONE,
C_SINCE, C_CREDIT, C_CREDIT_LIM
        , C_DISCOUNT, C_BALANCE, C_DATA
        ) ;

    COMMIT ;

END COMPOUND ;

```

```

}
else
{
    EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC

        SELECT W_STREET_1, W_STREET_2, W_CITY,
W_STATE, W_ZIP
        , D_STREET_1, D_STREET_2, D_CITY,
D_STATE, D_ZIP
        , C_LAST, C_FIRST, C_MIDDLE, C_STREET_1,
C_STREET_2
        , C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE,
C_CREDIT, C_CREDIT_LIM
        , C_DISCOUNT, C_BALANCE, C_DATA

    INTO :w_street_1 , :w_street_2 , :w_city , :w_state , :w_zip
        , :d_street_1 , :d_street_2 , :d_city , :d_state , :d_zip
        , :c_last , :c_first , :c_middle , :c_street_1 , :c_street_
2 , :c_city , :c_state
        , :c_zip , :c_phone , :c_since , :c_credit , :c_credit_li
m
        , :c_discount , :c_balance , :c_data :c_data_indicator

    FROM TABLE ( PAY_C_ID( :w_id
        , :d_id
        , :c_w_id
        , :c_d_id
        , :in_c_id
        , :h_date
        , :h_amount
        , :c_data_prefix_c_id
        )
        ) AS T( W_STREET_1, W_STREET_2,
W_CITY, W_STATE, W_ZIP
        , D_STREET_1, D_STREET_2, D_CITY,
D_STATE, D_ZIP
        , C_LAST, C_FIRST, C_MIDDLE,
C_STREET_1, C_STREET_2
        , C_CITY, C_STATE, C_ZIP, C_PHONE,
C_SINCE, C_CREDIT, C_CREDIT_LIM
        , C_DISCOUNT, C_BALANCE, C_DATA
        ) ;

    COMMIT ;

END COMPOUND ;

}

/* Update Output Structure with Timestamp */
payment->s_H_DATE_time = in_payment->s_H_DATE_time ;

#ifdef DEBUGIT
pay_debug(payment, in_payment, "Client after SQL call");
#endif /* DEBUGIT */

if ( sqlca.sqlcode != 0 )
{
    DLCHK( retry_tran ) ;
}

```

```

    sqlerror( PAYMENT_SQL , "PAY" , __FILE__ , __LINE__ ,
&sqlca );
    payment->s_transtatus = FATAL_SQLERROR ;
    clientRc = FATAL_SQLERROR ;

    pay_debug( payment, in_payment, "PAY failed" );

    EXEC SQL ROLLBACK WORK ;

    if ( sqlca.sqlcode != 0 )
    {
        sqlerror( PAYMENT_SQL , "ROLLBACK FAILED",
__FILE__ , __LINE__ , &sqlca );
    }

    return ( clientRc ) ;
}

// -----
// Order Status CLIENT
// -----

int ordstat_sql ( struct in_ordstat_struct * in_ordstat
, struct out_ordstat_struct * ordstat )
{
    struct sqlca sqlca ;

    EXEC SQL BEGIN DECLARE SECTION;

    struct vc_ord_in
    {
        short len ;
        char data[ 42 ] ;
    } * in_ord ;

    struct vc_ord_out
    {
        short len ;
        char data[ 446 ] ;
    } * out_ord ;

    EXEC SQL END DECLARE SECTION;

    int clientRc = TRAN_OK ;
    int itemIndex = 0 ;

    in_ord = (struct vc_ord_in *) in_ordstat ;
    in_ord->len = sizeof(struct in_ordstat_struct) -
SPGENERAL_ADJUST ;

    out_ord = (struct vc_ord_out *) ordstat ;
    out_ord->len = sizeof(struct out_ordstat_struct) -
SPGENERAL_ADJUST ;

#ifdef DEBUGIT
    ord_debug(ordstat, in_ordstat, "Client before SP call");
#endif /* DEBUGIT */

#ifdef SWAP_ENDIAN

```

```

        SWAP_BYTE(in_ordstat->s_C_ID);
        SWAP_BYTE(in_ordstat->s_W_ID);
        SWAP_BYTE(in_ordstat->s_D_ID);
#endif //SWAP_ENDIAN

    EXEC SQL CALL ords ( :*in_ord, :*out_ord ) ;

#ifdef SWAP_ENDIAN
    SWAP_BYTE(in_ordstat->s_C_ID);
    SWAP_BYTE(in_ordstat->s_W_ID);
    SWAP_BYTE(in_ordstat->s_D_ID);

    SWAP_BYTE(ordstat->s_C_BALANCE);
    SWAP_BYTE(ordstat->s_O_ENTRY_D_time);
    SWAP_BYTE(ordstat->s_C_ID);
    SWAP_BYTE(ordstat->s_O_ID);
    SWAP_BYTE(ordstat->s_O_CARRIER_ID);
    SWAP_BYTE(ordstat->s_ol_cnt);
    SWAP_BYTE(ordstat->s_transtatus);
    SWAP_BYTE(ordstat->deadlocks);
    for (itemIndex=0; itemIndex<ordstat->s_ol_cnt; itemIndex++)
    {
        SWAP_BYTE(ordstat-
>item[ itemIndex ].s_OL_DELIVERY_D_time);
        SWAP_BYTE(ordstat->item[ itemIndex ].s_OL_AMOUNT);
        SWAP_BYTE(ordstat->item[ itemIndex ].s_OL_I_ID);
        SWAP_BYTE(ordstat-
>item[ itemIndex ].s_OL_SUPPLY_W_ID);
        SWAP_BYTE(ordstat->item[ itemIndex ].s_OL_QUANTITY);
    }
#endif //SWAP_ENDIAN

    if ( sqlca.sqlcode == 0 )
    {
        // Propagate the field we already knew into the output
        structure
        // 60% of the time, we already new c_last (input c_id is 0)

        if ( in_ordstat->s_C_ID == 0 )
        {
            memcpy( ordstat->s_C_LAST , in_ordstat->s_C_LAST,
sizeof( ordstat->s_C_LAST ) ) ;
        }
        else
        {
            ordstat->s_C_ID = in_ordstat->s_C_ID ;
        }
    }
    else
    {
        sqlerror( ORDSTAT_SQL , "ORD" , __FILE__ , __LINE__ ,
&sqlca );
        ordstat->s_transtatus = FATAL_SQLERROR ;
        clientRc = FATAL_SQLERROR ;
    }

#ifdef DEBUGIT
    ord_debug(ordstat, in_ordstat, "Client after SP call");
#endif /* DEBUGIT */

    if ( ordstat->s_transtatus <= FATAL_SQLERROR )
    {

```

```

        ord_debug(ordstat, in_ordstat, "ORD failed");
        clientRc = FATAL_SQLERROR ;
    }

    return ( clientRc ) ;
}

// -----
// Delivery CLIENT
// -----

int delivery_sql ( struct in_delivery_struct * in_delivery
, struct out_delivery_struct * delivery )
{
    struct sqlca sqlca ;

    EXEC SQL BEGIN DECLARE SECTION;

    struct vc_del_in
    {
        short len ;
        char data[ 22 ] ;
    } * in_del ;

    struct vc_del_out
    {
        short len ;
        char data[ 50 ] ;
    } * out_del ;

    EXEC SQL END DECLARE SECTION;

    int clientRc = TRAN_OK ;
    int orderIndex = 0 ;

    /* Create Timestamp */

    in_delivery->s_O_DELIVERY_D_time = (sqlint64)
time( NULL ) ;

    in_del = (struct vc_del_in *) in_delivery ;
    in_del->len = sizeof(struct in_delivery_struct) -
SPGENERAL_ADJUST;

    out_del = (struct vc_del_out *) delivery ;
    out_del->len = sizeof(struct out_delivery_struct) -
SPGENERAL_ADJUST;

#ifdef DEBUGIT
    del_debug(delivery, in_delivery, "Client before SP call");
#endif /* DEBUGIT */

#ifdef SWAP_ENDIAN
    SWAP_BYTE(in_delivery->s_O_DELIVERY_D_time);
    SWAP_BYTE(in_delivery->s_W_ID);
    SWAP_BYTE(in_delivery->s_O_CARRIER_ID);
#endif //SWAP_ENDIAN

    EXEC SQL CALL dels ( :*in_del, :*out_del ) ;

#ifdef SWAP_ENDIAN
    SWAP_BYTE(in_delivery->s_O_DELIVERY_D_time);

```

```

SWAP_BYTE(in_delivery->s_W_ID);
SWAP_BYTE(in_delivery->s_O_CARRIER_ID);

for (orderIndex=0; orderIndex<10; orderIndex++) {
    SWAP_BYTE(delivery->s_O_ID[ orderIndex ]);
}
SWAP_BYTE(delivery->s_transtatus);
SWAP_BYTE(delivery->deadlocks);
#endif //SWAP_ENDIAN

#ifdef DEBUGIT
    del_debug(delivery, in_delivery, "Client after SP call");
#endif /* DEBUGIT */

    if ( sqlca.sqlcode != 0 )
    {
        sqlerror( DELIVERY_SQL, "DEL", __FILE__, __LINE__,
        &sqlca );
        delivery->s_transtatus = FATAL_SQLERROR ;
        clientRc = FATAL_SQLERROR ;
    }

    if ( delivery->s_transtatus <= FATAL_SQLERROR )
    {
        del_debug(delivery, in_delivery, "DEL failed");
        clientRc = FATAL_SQLERROR ;
    }

    return ( clientRc ) ;
}

// -----
// Stock CLIENT
// -----

#undef w_id
#undef d_id

int stocklev_sql ( struct in_stocklev_struct * in_stocklev
, struct out_stocklev_struct * stocklev )
{
    struct sqlca sqlca ;

    int clientRc = TRAN_OK ;

    EXEC SQL BEGIN DECLARE SECTION;

        // input

        sqlint32  threshold ;

        // output

        sqlint32  low_stock ;

    EXEC SQL END DECLARE SECTION;

    #define w_id  in_stocklev->s_W_ID
    #define d_id  in_stocklev->s_D_ID
    #define threshold in_stocklev->s_threshold
    #define low_stock stocklev->s_low_stock

```

```

stocklev->deadlocks = -1 ;
stocklev->s_transtatus = TRAN_OK ;

#ifdef DEBUGIT
    stk_debug(stocklev, in_stocklev, "Client before SQL call");
#endif /* DEBUGIT */

retry_tran:

    stocklev->deadlocks ++ ;

    EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC

        SELECT COUNT( S_I_ID ) INTO :low_stock

            FROM ( SELECT DISTINCT S_I_ID

                FROM ORDER_LINE , STOCK , DISTRICT

                WHERE D_W_ID = :w_id
                AND D_ID = :d_id
                AND OL_O_ID < d_next_o_id
                AND OL_O_ID >= ( d_next_o_id - 20 )
                AND OL_W_ID = D_W_ID
                AND OL_D_ID = D_ID
                AND S_I_ID = OL_I_ID
                AND S_W_ID = OL_W_ID
                AND S_QUANTITY < :threshold

            ) OLS

        WITH CS
        ;

    COMMIT ;

    END COMPOUND ;

#ifdef DEBUGIT
    stk_debug(stocklev, in_stocklev, "Client after SQL call");
#endif /* DEBUGIT */

    if ( sqlca.sqlcode != 0 )
    {
        DLCHK( retry_tran ) ;

        sqlerror( STOCKLEV_SQL , "STK" , __FILE__, __LINE__,
        &sqlca);
        stocklev->s_transtatus = FATAL_SQLERROR ;
        clientRc = FATAL_SQLERROR ;

        stk_debug( stocklev, in_stocklev, "STK failed" ) ;

        EXEC SQL ROLLBACK WORK ;

        if ( sqlca.sqlcode != 0 )
        {
            sqlerror( STOCKLEV_SQL, "ROLLBACK FAILED",
            __FILE__, __LINE__, &sqlca ) ;
        }
    }

```

```

return ( clientRc ) ;
}

Src.Common/Makefile

#####
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp.
## 1996 - 2006
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication
or
## disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
#####
#####

#
# Makefile - Makefile for Src.Common
#

!include $(TPCC_ROOT)/Makefile.config

#
#####
#####
# Preprocessor, Compiler and Linker Flags
#
#####
#####

BND_OPTS = GRANT PUBLIC \
            MESSAGES $.bnd.msg
PRP_OPTS = BINDFILE \
            OPTLEVEL 1 \
            ISOLATION RR \
            MESSAGES $.prep.msg \
            LEVEL $(TPCC_VERSION) \
            NOLINEMACRO

INCLUDES = -I$(TPCC_SQLLIB)$(SLASH)include -
            I$(TPCC_ROOT)$(SLASH)include

CFLAGS = $(CFLAGS_OS) $(CFLAGS_DEBUG)
$(INCLUDES) \
            -DSQLA_NOLINES -D$(DB2EDITION) -
            D$(DB2VERSION) \
            -D$(TPCC_SPTYPE)

UTIL_OBJ = tpccdbg$(OBJEXT) tpccctx$(OBJEXT)

#
#####
#####
# User Targets

```

```

#
#####
#####

all: connect $(UTIL_OBJ) disconnect

clean:
- $(ERASE) *$(OBJEXT) *.bnd *.msg tpcctx.c

#
#####
#####
# Helper Targets
#
#####
#####

connect:
- db2 connect to $(TPCC_DBNAME)

disconnect:
- db2 connect reset
- db2 terminate

rebind:
db2 bind tpcctx.bnd $(BND_OPTS)

#
#####
#####
# Build Rules
#
#####
#####

.SUFFIXES:
.SUFFIXES: $(OBJEXT) .c .sqc

.sqc.c:
@echo "Prepping $*.sqc"
-db2 prep $*.sqc $(PRP_OPTS)
@echo "Binding $*.bnd"
db2 bind $*.bnd $(BND_OPTS)

#
#####
#####
# Dependencies
#
#####
#####

# Source
tpccdbg$(OBJEXT): tpcdbg.c
tpccctx$(OBJEXT): tpcctx.c
tpccmisc$(OBJEXT): tpcmisc.c

# Headers
tpccdbg.c: $(TPCC_ROOT)/include/db2tpcc.h

```

Src.Common/tpcctx.sqc

```

/******
****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp.
1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
*****
****/

/*
 * tpcctx.sqc - TPC context code
 */

#include <stdlib.h>
#include <stdio.h>
#include <sqlutil.h>
#include "db2tpcc.h"

int connect_to_TM(char *in_dbname);
int connect_to_TM_auth(char *in_dbname, char *in_username,
char *in_password);
int disconnect_from_TM(void);
int create_context();
int destroy_context();
int attach_context(void*);
int detach_context(void*);
int get_context(void*);

int connect_to_TM(char *in_dbname)
{
return connect_to_TM_auth(in_dbname, "", "");
}

int connect_to_TM_auth(char *in_dbname, char *in_username,
char *in_password)
{
SQL_STRUCTURE sqlca sqlca;
int ConnectSQLCODE = 0;

EXEC SQL BEGIN DECLARE SECTION;
char dbname[9];
char username[129];
char password[15];
EXEC SQL END DECLARE SECTION;

SQLCODE = create_context();
if (SQLCODE != 0) { return SQLCODE; }

strncpy(dbname,in_dbname,8);
if (strcmp(in_username,"") == 0)
{
EXEC SQL CONNECT TO :dbname IN SHARE MODE;

```

```

} else {
strncpy(username,in_username,128);
strncpy(password,in_password,14);
EXEC SQL CONNECT TO :dbname IN SHARE MODE
USER :username USING :password;
}

ConnectSQLCODE = SQLCODE;
if (ConnectSQLCODE != 0)
{
sqlerror( CLIENT_SQL, "CONNECT", __FILE__, __LINE__,
&sqlca);

SQLCODE = destroy_context();
if (SQLCODE != 0) { return SQLCODE; }

return ConnectSQLCODE;
}

return 0;
}

int disconnect_from_TM(void)
{
SQL_STRUCTURE sqlca sqlca;
int DisconnectSQLCODE = 0;

EXEC SQL CONNECT RESET;

DisconnectSQLCODE = SQLCODE;
if (DisconnectSQLCODE != 0) {
sqlerror( CLIENT_SQL, "DISCONNECT", __FILE__,
__LINE__, &sqlca);
}

SQLCODE = destroy_context();
if (SQLCODE != 0) { return SQLCODE; }

if (DisconnectSQLCODE) {
return DisconnectSQLCODE;
}
return 0;
}

int create_context(void)
{
SQL_STRUCTURE sqlca sqlca;
void *ctx;

sqlcSetTypeCtx(SQL_CTX_MULTI_MANUAL);
sqlcBeginCtx(&ctx, SQL_CTX_BEGIN_ALL, NULL, &sqlca);

if (SQLCODE != 0) {
sqlerror( CLIENT_SQL, "CREATE", __FILE__, __LINE__,
&sqlca);
return SQLCODE;
}

return 0;
}

int attach_context(void *ctx)

```

```

{
    SQL_STRUCTURE sqlca sqlca;

    sqlcAttachToCtx(ctx, NULL, &sqlca);

    if (SQLCODE != 0) {
        sqlerror( CLIENT_SQL, "ATTACH", __FILE__, __LINE__,
        &sqlca);
        return SQLCODE;
    }

    return 0;
}

int detach_context(void *ctx)
{
    SQL_STRUCTURE sqlca sqlca;

    sqlcDetachFromCtx(ctx, NULL, &sqlca);

    if (SQLCODE != 0) {
        sqlerror( CLIENT_SQL, "DETACH", __FILE__, __LINE__,
        &sqlca);
        return SQLCODE;
    }

    return 0;
}

int destroy_context(void)
{
    SQL_STRUCTURE sqlca sqlca;
    void *ctx;

    SQLCODE = get_context(&ctx);
    if (SQLCODE) { return SQLCODE; }

    sqlcEndCtx(&ctx, SQL_CTX_END_ALL, NULL, &sqlca);

    if (SQLCODE != 0) {
        sqlerror( CLIENT_SQL, "DESTROY", __FILE__, __LINE__,
        &sqlca);
        return SQLCODE;
    }

    return 0;
}

int get_context(void **ctx)
{
    SQL_STRUCTURE sqlca sqlca;

    sqlcGetCurrentCtx(ctx, NULL, &sqlca);

    if (SQLCODE != 0) {
        sqlerror( CLIENT_SQL, "GETCTX", __FILE__, __LINE__,
        &sqlca);
        return SQLCODE;
    }

    return 0;
}

```

Src.Common/tpccdbg.c

```

/*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp.
** 1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
** IBM Corp.
*****/

/*
 * tccdbg.c - Debugging Routines
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <time.h>

#include "sqlca.h"
#include "sql.h"
#include "db2tpcc.h"
#include "tpccdbg.h"

#define DEBUG_FILENAME_SZ 128
#define DEBUG_PATH_SIZE 128

void del_print();
void new_print();
void ord_print();
void pay_print();
void stk_print();

void current_tmstamp(char *buf);

static int debugInit = 0;
static char debugPath[DEBUG_PATH_SIZE] = "";

/*-----*/
/*      InitializeDebug      */
/*-----*/
__inline void InitializeDebug(void) {
    if (debugInit == 0) {
        char *p = getenv("TPCC_DEBUGDIR");
        if (p) {
            strncpy(debugPath, p, DEBUG_PATH_SIZE);
        } else {
            strcpy(debugPath, "C:\\temp");
        }
    }
}

```

```

    strcat(debugPath, "\\");
}
debugInit = 1;
}

/*-----*/
/*      sqlerror      */
/*-----*/
void sqlerror(int tranType, char *msg, char *file, int line,
SQL_STRUCTURE sqlca *psqlca)
{
    FILE *err_fp = NULL;
    char err_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];
    char tranName[16];
    int j,k;
    char timeStamp[27];
    char errStr[512] = "";

    InitializeDebug();
    strncpy(err_fn, debugPath, DEBUG_PATH_SIZE);
    current_tmstamp(&timeStamp[0]);
    timeStamp[19] = (char)NULL;

    switch(tranType)
    {
        case NEWORD_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "new.err.out");
            strcpy(tranName, "NEW_ORDER");
            break;

        case DELIVERY_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "del.err.out");
            strcpy(tranName, "DELIVERY");
            break;

        case PAYMENT_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "pay.err.out");
            strcpy(tranName, "PAYMENT");
            break;

        case ORDSTAT_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "ord.err.out");
            strcpy(tranName, "ORDER_STAT");
            break;

        case STOCKLEV_SQL:
            //sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "stk.err.out");
            strcpy(tranName, "STOCK_LVL");
            break;

        case 0:
            strcat(err_fn, "cli.err.out");
            strcpy(tranName, "CLIENT");
            break;

        default:
            return;
    }
}

```

```

}

/* Generate Formatted Error Message */
sqlaintp(terrStr, 512, 78, psqlca);

err_fp = fopen(err_fn, "a+");

fprintf(err_fp, "-----\n");
fprintf(err_fp, "Transaction: %s (%s)\n", tranName, msg);
fprintf(err_fp, "FILE %s (%u)\n", file, line);
fprintf(err_fp, "SQLCODE %d", psqlca->sqlcode);
fprintf(err_fp, "TIME %s\n", timeStamp);
fprintf(err_fp, "-----\n");
fprintf(err_fp, "%s", errStr);
fprintf(err_fp, "-----\n");

if (psqlca->sqlerrmc[0] != ' ' || psqlca->sqlerrmc[1] != ' ')
{
    fprintf(err_fp, "slerrmc: ");

    for(j = 0; j < 5; j++)
    {
        for(k = 0; k < 16; k++) {
            int pos = j * 16 + k;
            if (pos < 70) fprintf(err_fp, "%02x ", psqlca-
->sqlerrmc[pos]);
            else fprintf(err_fp, " ");
        }
        fprintf(err_fp, " |");
        for(k = 0; k < 16; k++) {
            int pos = j * 16 + k;
            char c = ' ';
            if (pos < 70) {
                c = psqlca->sqlerrmc[pos];
                if (!isprint(c)) c = ' ';
            }
            fprintf(err_fp, "%c", c);
        }
        fprintf(err_fp, "\n");
        if (j < 4) fprintf(err_fp, " ");
    }
}

fprintf(err_fp, "sqlerrp: ");
for(j = 0; j < 8; j++)
    fprintf(err_fp, "%c", psqlca->sqlerrp[j]);
fprintf(err_fp, "\n");

fprintf(err_fp, "sqlerrd: ");
for(j = 0; j < 6; j++)
    fprintf(err_fp, "%d", psqlca->sqlerrd[j]);
fprintf(err_fp, "\n");

if (psqlca->sqlwarn[0] != ' ')
{
    fprintf(err_fp, "sqlwarn: ");
    for(j = 0; j < 8; j++)
        fprintf(err_fp, "%c ", psqlca->sqlwarn[j]);
    fprintf(err_fp, "\n");
}

```

```

fprintf(err_fp, "\n");
fclose(err_fp);
}

/*-----*/
/* del_debug */
/*-----*/
void del_debug (struct out_delivery_struct *delivery_ptr,
                struct in_delivery_struct *in_delivery,
                char *msg)
{
    char debug_fn[DEBUG_PATH_SIZE +
DEBUG_FILENAME_SZ];

    InitializeDebug();
    strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
    strcat(debug_fn, "del.debug.out");
    del_print(delivery_ptr, in_delivery, debug_fn, msg);
}

/*-----*/
/* del_print */
/*-----*/
void del_print (struct out_delivery_struct *delivery_ptr,
                struct in_delivery_struct *in_delivery,
                char *filename,
                char *msg)
{
    FILE *debug_fp;
    char timeStamp[27];
    int j;

    current_tmstamp(&timeStamp[0]);
    timeStamp[19] = (char)NULL;

    if ((debug_fp = fopen(filename, "a+")) == NULL)
    {
        return;
    }

    fprintf(debug_fp, "Delivery debug information follows %s
(%s)\n", timeStamp, msg);

    fprintf(debug_fp, "\n=====
=====");

    fprintf(debug_fp, "in_delivery_struct {\n");
    fprintf(debug_fp, "ts_W_ID = %d (%X)\n",
            in_delivery->s_W_ID, in_delivery->s_W_ID);
    fprintf(debug_fp, "ts_O_CARRIER_ID = %d (%X)\n",
            in_delivery->s_O_CARRIER_ID, in_delivery-
>s_O_CARRIER_ID);
    fprintf(debug_fp, "ts_O_DELIVERY_D = %ld (%lX)\n",
            in_delivery->s_O_DELIVERY_D_time, in_delivery-
>s_O_DELIVERY_D_time);
    fprintf(debug_fp, "}\n\n");

    fprintf(debug_fp, "out_delivery_struct {\n");
    fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
            delivery_ptr->s_transtatus, delivery_ptr->s_transtatus);
}

```

```

fprintf(debug_fp, "\tdeadlocks = %d (%X)\n",
        delivery_ptr->deadlocks, delivery_ptr->deadlocks);

for (j = 0; j < 10; j++) {
    fprintf(debug_fp, "\tts_O_ID[%d] = %d\n",
            j, delivery_ptr->s_O_ID[j]);
}
fprintf(debug_fp, "\t}\n\n");
fclose(debug_fp);
}

/*-----*/
/* new_debug */
/*-----*/
void new_debug (struct out_neword_struct *neword_ptr,
                struct in_neword_struct *in_neword,
                char *msg)
{
    char debug_fn[DEBUG_PATH_SIZE +
DEBUG_FILENAME_SZ];

    InitializeDebug();
    strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
    strcat(debug_fn, "new.debug.out");
    new_print(neword_ptr, in_neword, debug_fn, msg);
}

/*-----*/
/* new_print */
/*-----*/
void new_print (struct out_neword_struct *neword_ptr,
                struct in_neword_struct *in_neword,
                char *filename,
                char *msg)
{
    FILE *debug_fp;
    char timeStamp[27];
    int j, items;

    current_tmstamp(&timeStamp[0]);
    timeStamp[19] = (char)NULL;

    if ((debug_fp = fopen(filename, "a+")) == NULL)
    {
        return;
    }

    fprintf(debug_fp, "New order debug information follows %s
(%s)\n", timeStamp, msg);

    fprintf(debug_fp, "\n=====
=====");

    fprintf(debug_fp, "in_neword_struct {\n");

    fprintf(debug_fp, "ts_C_ID = %d (%X)\n",
            in_neword->s_C_ID, in_neword->s_C_ID);
    fprintf(debug_fp, "ts_W_ID = %d (%X)\n",
            in_neword->s_W_ID, in_neword->s_W_ID);
    fprintf(debug_fp, "ts_D_ID = %d (%X)\n",
            in_neword->s_D_ID, in_neword->s_D_ID);
}

```

```

fprintf(debug_fp, "ts_O_OL_CNT = %d (%X)\n",
        in_neword->s_O_OL_CNT, in_neword->s_O_OL_CNT);
fprintf(debug_fp, "ts_all_local = %d (%X)\n",
        in_neword->s_all_local, in_neword->s_all_local);
fprintf(debug_fp, "ts_O_ENTRY_D = %lld (%lX)\n",
        in_neword->s_O_ENTRY_D_time, in_neword-
>s_O_ENTRY_D_time);
// fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
//        in_neword->s_transtatus, in_neword->s_transtatus);
// fprintf(debug_fp, "tduplicate_items= %d (%X)\n",
//        in_neword->duplicate_items, in_neword-
>duplicate_items);

fprintf(debug_fp, "titems {n}");
items = in_neword->s_O_OL_CNT;
for (j=0; j<items; j++) {
    if(j != 0)
        fprintf(debug_fp, "\n");
    fprintf(debug_fp, "\t\tts_OL_I_ID[%d] = %d (%X)\n",
            j, in_neword->in_item[j].s_OL_I_ID, in_neword-
>in_item[j].s_OL_I_ID);
    fprintf(debug_fp, "\t\tts_OL_SUPPLY_W_ID[%d] = %d
(%X)\n",
            j, in_neword->in_item[j].s_OL_SUPPLY_W_ID,
in_neword->in_item[j].s_OL_SUPPLY_W_ID);
    fprintf(debug_fp, "\t\tts_OL_QUANTITY[%d] = %d (%X)\n",
            j, in_neword->in_item[j].s_OL_QUANTITY, in_neword-
>in_item[j].s_OL_QUANTITY);
}
fprintf(debug_fp, "\t}\n\n");

fprintf(debug_fp, "out_neword_struct {n}");
fprintf(debug_fp, "ts_C_LAST = %s\n",
        neword_ptr->s_C_LAST);
fprintf(debug_fp, "ts_C_CREDIT = %s\n",
        neword_ptr->s_C_CREDIT);
fprintf(debug_fp, "ts_W_TAX = %d\n",
        neword_ptr->s_W_TAX);
fprintf(debug_fp, "ts_D_TAX = %d\n",
        neword_ptr->s_D_TAX);
fprintf(debug_fp, "ts_C_DISCOUNT = %d\n",
        neword_ptr->s_C_DISCOUNT);
fprintf(debug_fp, "ts_O_ID = %d (%X)\n",
        neword_ptr->s_O_ID, neword_ptr->s_O_ID);
fprintf(debug_fp, "ts_O_OL_CNT = %d (%X)\n",
        neword_ptr->s_O_OL_CNT, neword_ptr-
>s_O_OL_CNT);
fprintf(debug_fp, "ts_O_ENTRY_D = %lld (%lX)\n",
        neword_ptr->s_O_ENTRY_D_time, neword_ptr-
>s_O_ENTRY_D_time);
fprintf(debug_fp, "ts_total_amount = %d\n",
        neword_ptr->s_total_amount);
fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
        neword_ptr->s_transtatus, neword_ptr->s_transtatus);
fprintf(debug_fp, "tdeadlocks = %d (%X)\n",
        neword_ptr->deadlocks, neword_ptr->deadlocks);

// fprintf(debug_fp, "ts_W_ID = %d (%X)\n",
//        neword_ptr->s_W_ID, neword_ptr->s_W_ID);
// fprintf(debug_fp, "ts_D_ID = %d (%X)\n",
//        neword_ptr->s_D_ID, neword_ptr->s_D_ID);
// fprintf(debug_fp, "ts_all_local = %d (%X)\n",

```

```

//        neword_ptr->s_all_local, neword_ptr->s_all_local);
// fprintf(debug_fp, "tduplicate_items= %d (%X)\n",
//        neword_ptr->duplicate_items, neword_ptr-
>duplicate_items);

fprintf(debug_fp, "titems {n}");
items = neword_ptr->s_O_OL_CNT;
for (j=0; j<items; j++) {
    if(j != 0)
        fprintf(debug_fp, "\n");
    fprintf(debug_fp, "\t\tts_I_NAME[%d] = %s\n",
            j, neword_ptr->item[j].s_I_NAME);
    fprintf(debug_fp, "\t\tts_I_PRICE[%d] = %d\n",
            j, neword_ptr->item[j].s_I_PRICE);
    fprintf(debug_fp, "\t\tts_OL_AMOUNT[%d] = %d\n",
            j, neword_ptr->item[j].s_OL_AMOUNT);
    fprintf(debug_fp, "\t\tts_S_QUANTITY[%d] = %d (%X)\n",
            j, neword_ptr->item[j].s_S_QUANTITY, neword_ptr-
>item[j].s_S_QUANTITY);
    fprintf(debug_fp, "\t\tts_brand_generic[%d] = %c\n",
            j, neword_ptr->item[j].s_brand_generic);
}
fprintf(debug_fp, "\t}\n\n");
fclose(debug_fp);

/*-----*/
/* ord_debug */
/*-----*/
void ord_debug (struct out_ordstat_struct *ordstat_ptr,
                struct in_ordstat_struct *in_ordstat,
                char *msg)
{
    char debug_fn[DEBUG_PATH_SIZE +
DEBUG_FILENAME_SZ];

    InitializeDebug();
    strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
    strcat(debug_fn, "ord.debug.out");
    ord_print(ordstat_ptr, in_ordstat, debug_fn, msg);
}

/*-----*/
/* ord_print */
/*-----*/
void ord_print (struct out_ordstat_struct *ordstat_ptr,
                struct in_ordstat_struct *in_ordstat,
                char *filename,
                char *msg)
{
    FILE *debug_fp;
    char timeStamp[27];
    int j, items;

    current_tmstamp(&timeStamp[0]);
    timeStamp[19] = (char)NULL;

    if ((debug_fp = fopen(filename, "a+")) == NULL)
    {

```

```

return;
}

fprintf(debug_fp, "Order status debug information follows %s
(%s)\n", timeStamp, msg);

fprintf(debug_fp, "\n=====
=====");

fprintf(debug_fp, "in_ordstat_struct {n}");
fprintf(debug_fp, "ts_W_ID = %d (%X)\n",
        in_ordstat->s_W_ID, in_ordstat->s_W_ID);
fprintf(debug_fp, "ts_D_ID = %d (%X)\n",
        in_ordstat->s_D_ID, in_ordstat->s_D_ID);
fprintf(debug_fp, "ts_C_ID = %d (%X)\n",
        in_ordstat->s_C_ID, in_ordstat->s_C_ID);
fprintf(debug_fp, "ts_C_LAST = %s\n",
        in_ordstat->s_C_LAST);
fprintf(debug_fp, ")\n");

fprintf(debug_fp, "out_ordstat_struct {n}");
fprintf(debug_fp, "ts_C_ID = %d (%X)\n",
        ordstat_ptr->s_C_ID, ordstat_ptr->s_C_ID);
fprintf(debug_fp, "ts_C_FIRST = %s\n",
        ordstat_ptr->s_C_FIRST);
fprintf(debug_fp, "ts_C_MIDDLE = %s\n",
        ordstat_ptr->s_C_MIDDLE);
fprintf(debug_fp, "ts_C_LAST = %s\n",
        ordstat_ptr->s_C_LAST);
fprintf(debug_fp, "ts_C_BALANCE = %lld\n",
        ordstat_ptr->s_C_BALANCE);
fprintf(debug_fp, "ts_O_ID = %d (%X)\n",
        ordstat_ptr->s_O_ID, ordstat_ptr->s_O_ID);
fprintf(debug_fp, "ts_O_ENTRY_D = %lld (%lX)\n",
        ordstat_ptr->s_O_ENTRY_D_time, ordstat_ptr-
>s_O_ENTRY_D_time);
fprintf(debug_fp, "ts_O_CARRIER_ID = %d (%X)\n",
        ordstat_ptr->s_O_CARRIER_ID, ordstat_ptr-
>s_O_CARRIER_ID);
fprintf(debug_fp, "ts_ol_cnt = %d (%X)\n",
        ordstat_ptr->s_ol_cnt, ordstat_ptr->s_ol_cnt);
fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
        ordstat_ptr->s_transtatus, ordstat_ptr->s_transtatus);
fprintf(debug_fp, "tdeadlocks = %d (%X)\n",
        ordstat_ptr->deadlocks, ordstat_ptr->deadlocks);

fprintf(debug_fp, "titems {n}");
items = ordstat_ptr->s_ol_cnt;
for (j = 0; j < items; j++) {
    if(j != 0)
        fprintf(debug_fp, "\n");
    fprintf(debug_fp, "\t\tts_OL_SUPPLY_W_ID[%d] = %d
(%X)\n",
            j, ordstat_ptr->item[j].s_OL_SUPPLY_W_ID,
ordstat_ptr->item[j].s_OL_SUPPLY_W_ID);
    fprintf(debug_fp, "\t\tts_OL_I_ID[%d] = %d (%X)\n",
            j, ordstat_ptr->item[j].s_OL_I_ID, ordstat_ptr-
>item[j].s_OL_I_ID);
    fprintf(debug_fp, "\t\tts_OL_QUANTITY[%d] = %d (%X)\n",
            j, ordstat_ptr->item[j].s_OL_QUANTITY, ordstat_ptr-
>item[j].s_OL_QUANTITY);
    fprintf(debug_fp, "\t\tts_OL_AMOUNT[%d] = %d\n",

```



```

        in_stocklev->s_threshold, in_stocklev->s_threshold);
fprintf(debug_fp, "\n\n");

fprintf(debug_fp, "out_stocklev_struct {\n");
fprintf(debug_fp, "\tts_transtatus = %d (%X)\n",
        stocklev->s_transtatus, stocklev->s_transtatus);
fprintf(debug_fp, "\tdeadlocks = %d (%X)\n",
        stocklev->deadlocks, stocklev->deadlocks);
fprintf(debug_fp, "\ts_low_stock = %d (%X)\n",
        stocklev->s_low_stock, stocklev->s_low_stock);
fprintf(debug_fp, "\n\n");
fclose(debug_fp);
}

void current_tmstamp(char *buf)
{
    time_t t = time(NULL);
    strncpy(buf, ctime(&t), 19);
}

include/db2tpcc.h

/*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp.
1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
*****/

/*
* db2tpcc.h - Macros and Miscellany
*/

#ifndef __DB2TPCC_H
#define __DB2TPCC_H

#include <sys/types.h>
typedef __int16 int16_t;
typedef __int32 int32_t;
typedef __int64 int64_t;

#include "lval.h"

/*
*****/
/* Transaction Return Codes (s_transtatus)
*/
/*****/
*/

```

```

#define INVALID_ITEM      100
#define TRAN_OK          0
#define FATAL_SQLERROR   -1

/*
*****
*/
/* Definition of Unused and Bad Items */
/*
*****
*/
/* Define unused item ID to be 0. This allows the SUT to
determine the */
/* number of items in the order as required by 2.4.1.3 and
2.4.2.2 since */
/* the assumption that any item with OL_I_ID = 0 is unused will
be true. */
/* This in turn requires that the value used for an invalid item is
*/
/* equal to ITEMS + 1. */
/*
*****
*/

#define INVALID_ITEM_ID (2 * ITEMS) + 1
#define UNUSED_ITEM_ID 0

#define MIN_WAREHOUSE 1
#define MAX_WAREHOUSE WAREHOUSES

/*****
*****/
/* NURand Constants */
/* C_C_LAST_RUN and C_C_LAST_LOAD must adhere to
clause 2.1.6. */
/* Analysis indicates that a C_LAST delta of 85 is optimal.
*/
/*****
*****/

#define C_C_LAST_RUN 88
#define C_C_LAST_LOAD 173
#define C_C_ID 319
#define C_OL_I_ID 3849
#define A_C_LAST 255
#define A_C_ID 1023
#define A_OL_I_ID 8191

/*****
*****/
/* Transaction Type Identifiers */
/*****
*****/

#define CLIENT_SQL 0
#define NEWORD_SQL 1
#define PAYMENT_SQL 2
#define ORDSTAT_SQL 3
#define DELIVERY_SQL 4
#define STOCKLEV_SQL 5

#define SPGENERAL_PAD 3

```

```

#define SPGENERAL_ADJUST sizeof(int16_t)

struct in_neword_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    struct in_items_struct {
        int32_t s_OL_I_ID;
        int32_t s_OL_SUPPLY_W_ID;
        int16_t s_OL_QUANTITY;
        int16_t pad1[3];
    } in_item[15];
    int64_t s_O_ENTRY_D_time; /* init by SUT */
    int32_t s_C_ID;
    int32_t s_W_ID;
    int16_t s_D_ID;
    int16_t s_O_OL_CNT; /* init by SUT */
    int16_t s_all_local;
    int16_t duplicate_items;
};

struct out_neword_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    struct items_struct {
        int32_t s_I_PRICE;
        int32_t s_OL_AMOUNT;
        int16_t s_S_QUANTITY;
        int16_t pad2;
        char s_I_NAME[25];
        char s_brand_generic;
    } item[15];
    int64_t s_O_ENTRY_D_time;
    int32_t s_W_TAX;
    int32_t s_D_TAX;
    int32_t s_C_DISCOUNT;
    int32_t s_total_amount;
    int32_t s_O_ID;
    int16_t s_O_OL_CNT;
    int16_t s_transtatus;
    int16_t deadlocks;
    char s_C_LAST[17];
    char s_C_CREDIT[3];
};

struct in_payment_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int64_t s_H_DATE_time; /* init by SUT */
    int64_t s_H_AMOUNT;
    int32_t s_W_ID;
    int32_t s_C_W_ID;
    int32_t s_C_ID;
    int16_t s_C_D_ID;
    int16_t s_D_ID;
    char s_C_LAST[17];
};

struct out_payment_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int64_t s_H_DATE_time;
    int64_t s_C_SINCE_time;
};

```

```

int64_t s_C_CREDIT_LIM;
int64_t s_C_BALANCE;
int32_t s_C_DISCOUNT;
int32_t s_C_ID;
int16_t s_transtatus;
int16_t deadlocks;
char s_W_STREET_1[21];
char s_W_STREET_2[21];
char s_W_CITY[21];
char s_W_STATE[3];
char s_W_ZIP[10];
char s_D_STREET_1[21];
char s_D_STREET_2[21];
char s_D_CITY[21];
char s_D_STATE[3];
char s_D_ZIP[10];
char s_C_FIRST[17];
char s_C_MIDDLE[3];
char s_C_LAST[17];
char s_C_STREET_1[21];
char s_C_STREET_2[21];
char s_C_CITY[21];
char s_C_STATE[3];
char s_C_ZIP[10];
char s_C_PHONE[17];
char s_C_CREDIT[3];
char s_C_DATA[201];
};

struct in_ordstat_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int32_t s_C_ID;
int32_t s_W_ID;
int16_t s_D_ID;
int16_t pad1[3];
char s_C_LAST[17];
};

struct out_ordstat_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int64_t s_C_BALANCE;
int64_t s_O_ENTRY_D_time;
int32_t s_C_ID;
int32_t s_O_ID;
int16_t s_O_CARRIER_ID;
int16_t s_ol_cnt;
int16_t pad1[2];
struct oitems_struct {
int64_t s_OL_DELIVERY_D_time;
int32_t s_OL_AMOUNT;
int32_t s_OL_I_ID;
int32_t s_OL_SUPPLY_W_ID;
int16_t s_OL_QUANTITY;
int16_t pad2;
} item[15];
int16_t s_transtatus;
int16_t deadlocks;
char s_C_FIRST[17];
char s_C_MIDDLE[3];
char s_C_LAST[17];
};

```

```

};

struct in_delivery_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int64_t s_O_DELIVERY_D_time; /* init by SUT */
int32_t s_W_ID;
int16_t s_O_CARRIER_ID;
};

struct out_delivery_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int32_t s_O_ID[10];
int16_t s_transtatus;
int16_t deadlocks;
};

struct in_stocklev_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int32_t s_threshold;
int32_t s_W_ID;
int16_t s_D_ID;
};

struct out_stocklev_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int32_t s_low_stock;
int16_t s_transtatus;
int16_t deadlocks;
};

/*
*****
*/
/* Transaction Prototypes */
/*
*****
*/

#ifdef __cplusplus
extern "C" {
#endif

extern int neword_sql(struct in_neword_struct*, struct
out_neword_struct*);
extern int payment_sql(struct in_payment_struct*, struct
out_payment_struct*);
extern int ordstat_sql(struct in_ordstat_struct*, struct
out_ordstat_struct*);
extern int delivery_sql(struct in_delivery_struct*, struct
out_delivery_struct*);
extern int stocklev_sql(struct in_stocklev_struct*, struct
out_stocklev_struct*);

#ifdef __cplusplus
}
#endif

```

```

/*
*****
*/
/* DB2 Connect/Disconnect & Thread Context Wrappers */
/*
/*
*****
*/

#ifdef __cplusplus
extern "C" {
#endif

extern int connect_to_TM(char*);
extern int connect_to_TM_auth(char*, char*, char*);
extern int disconnect_from_TM(void);

extern int create_context(void);
extern int destroy_context(void);
extern int get_context(void**);
extern int attach_context(void*);
extern int detach_context(void*);

#ifdef __cplusplus
}
#endif

#endif // __DB2TPCC_H

include/lval.h

#ifdef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 81600
#define DISTRICTS_PER_WAREHOUSE 10
#define CUSTOMERS_PER_DISTRICT 3000
#define ITEMS 100000
#define STOCK_PER_WAREHOUSE 100000
#define MIN_OL_PER_ORDER 5
#define MAX_OL_PER_ORDER 15
#define NU_ORDERS_PER_DISTRICT 900
#endif // __LVAL_H

include/tpccapp.h

/*****
****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp.
1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
*****/

```

```

/*
 * tpcapp.h - Application Macros
 */

#ifndef __TPCCAPP_H
#define __TPCCAPP_H

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#include "sqlenv.h"
#define daricall __stdcall

#include "sqlca.h"
#include "sqlcodes.h"

#ifdef SWAP_ENDIAN
#define SWAP_BYTE(Var) SwapEndian((void*)&Var,
sizeof(Var))

/*****
FUNCTION: SwapEndian
PURPOSE: Swap the byte order of a structure
EXAMPLE: int I=0x12345678; SWAP_BYTE(I); I =>
0x78563412;
IMPLEMENTATION: Fold Addr in half, swap header & tail by
XOR op
e.g.: *a = 0x12 [ Addr + 0];
      *b = 0x78 [ Add + 4 - 0 - 1 = Addr+3];
      *a ^= *b; // sets *a to 0x6A
      *b ^= *a; // sets *b to 0x12
      *a ^= *b; // sets *a to 0x78

      Now *a => 0x78 && *b => 0x12
*****/

void SwapEndian(void *Addr, int nb)
{
    int i;
    for (i=0; i<nb/2; i++)
    {
        char *a = (char*)Addr+i;
        char *b = (char*)Addr+(nb-i-1);

        *a ^= *b;
        *b ^= *a;
        *a ^= *b;
    }
}
#endif //SWAP_ENDIAN

/*****
***
*/
SQLCODE Macros
/*****
***

```

```

#define DLCHK(a) \
    if (sqlca.sqlcode == SQL_RC_E911) { goto a; }

/*
*****
*/
/* In NOT ATOMIC COMPOUND SQL, all statements will be
executed, but not */
/* all will necessarily complete successfully. We can use
sqlerrd(4) to */
/* determine how many statements succeeded, but this won't tell
us what */
/* statements failed. In order to determine this, we need to look
at */
/* sqlerrmc, which has the following structure:
HHHXNNNSSSSXNNNSSSS... */
/* (See the docs for more details.) Since we're interested in the
first */
/* failing statement, we can look at elements 5 and 6, which will
contain */
/* the first two digits of NNN (which is right-padded with spaces).
We */
/* need to look at the first two digits since some of our
compound blocks */
/* have > 9 statements. We convert these digits from ASCII to
an int and */
/* set 'last' to this value. */
/*
*****
*/

#define NACOMPCHK(last) \
    if (sqlca.sqlcode != SQL_RC_E1339) { last = -1; } \
    else { int a = ((sqlca.sqlerrmc[4] == 0x20) ? 0 : \
sqlca.sqlerrmc[4]-0x30); \
          int b = ((sqlca.sqlerrmc[5] == 0x20) ? 0 : sqlca.sqlerrmc[5]-\
0x30); \
          if (b == 0) { last = a; } else { last = a * 10 + b; } \
    }

#endif // __TPCCAPP_H

include/tpccdbg.h

/*****
****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp.
1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
*****
***

```

```

/*
 * tpccdbg.h - Debugging Macros
 */

#ifndef __TPCCDBG_H
#define __TPCCDBG_H

#ifdef __cplusplus
extern "C" {
#endif

extern void new_debug (struct out_neword_struct *neword_ptr,
                      struct in_neword_struct *in_neword_ptr,
                      char *msg);
extern void pay_debug (struct out_payment_struct
*payment_ptr,
                      struct in_payment_struct *in_payment_ptr,
                      char *msg);
extern void ord_debug (struct out_ordstat_struct *ordstat_ptr,
                      struct in_ordstat_struct *in_ordstat_ptr,
                      char *msg);
extern void del_debug (struct out_delivery_struct *delivery_ptr,
                      struct in_delivery_struct *in_delivery_ptr,
                      char *msg);
extern void stk_debug (struct out_stocklev_struct *stocklev_ptr,
                      struct in_stocklev_struct *in_stocklev_ptr,
                      char *msg);

extern void new_print (struct out_neword_struct *neword_ptr,
                      struct in_neword_struct *in_neword_ptr,
                      char *filename,
                      char *msg);
extern void pay_print (struct out_payment_struct *payment_ptr,
                      struct in_payment_struct *in_payment_ptr,
                      char *filename,
                      char *msg);
extern void ord_print (struct out_ordstat_struct *ordstat_ptr,
                      struct in_ordstat_struct *in_ordstat_ptr,
                      char *filename,
                      char *msg);
extern void del_print (struct out_delivery_struct *delivery_ptr,
                      struct in_delivery_struct *in_delivery_ptr,
                      char *filename,
                      char *msg);
extern void stk_print (struct out_stocklev_struct *stocklev_ptr,
                      struct in_stocklev_struct *in_stocklev_ptr,
                      char *filename,
                      char *msg);

#ifdef __cplusplus
}
#endif

#endif // __TPCCDBG_H

tpccenv.bat

@REM
*****
*

```

```

@REM Licensed Materials - Property of IBM
@REM
@REM Governed under the terms of the International
@REM License Agreement for Non-Warranted Sample Code.
@REM
@REM (C) COPYRIGHT International Business Machines
@REM Corp. 1996 - 2006
@REM All Rights Reserved.
@REM
@REM US Government Users Restricted Rights - Use,
@REM duplication or
@REM disclosure restricted by GSA ADP Schedule Contract
@REM with IBM Corp.
@REM
*****
**

@REM
@REM tpccenv.bat - Windows Environment Setup
@REM

@REM The Kit Version
set TPCC_VERSION=CK040318

@REM The DB2 Instance Name (for DB2)
set DB2INSTANCE=%USERNAME%

@REM The OS being used (i.e. "UNIX", "WINDOWS")
set PLATFORM=WINDOWS

@REM The type of make command and slash used by the OS
@REM (i.e. UNIX - "/", WINDOWS - "\")
@REM These are referenced all over the kit.
set SLASH=
set MAKE=nmake

@REM Specifies whether or not to use dari stored proc's for the
@REM TPC-C driver. Set to either DARIVERSION or NONDARI;
@REM set TPCC_SPTYPE=NOSP
@REM set TPCC_SPTYPE=SPGENERAL2
@REM set TPCC_SPTYPE=SPGENERAL
@REM set TPCC_SPTYPE=DARI2SQLDA

set DB2VERSION=v8

@REM The schema name is typically the SQL authorization ID
@REM (or username).
@REM This is required for runstats and EEE.
set TPCC_SCHEMA=%USERNAME%

@REM DB2 EE/EEE Configuration
set DB2EDITION=EE
@REM set DB2EDITION=EEE
set DB2NODE=0
@REM set to the number of nodes you have. Set to 1 for EE.
set DB2NODES=1

@REM TPCC General Configuration
@REM ** IMPORTANT NOTE **
@REM The kit is not guaranteed to work properly if
TPCC_ROOT or TPCC_SQLLIB

```

```

@REM have spaces in them. If you absolutely must use paths
@REM with spaces,
@REM then the entire path must be surrounded by double
@REM quotes.
@REM For example: HOME="C:\Program Files\IBM"
set HOME=C:\home\tpcc
set TPCC_DBNAME=TPCC
set TPCC_ROOT=%HOME%\tpc-c.ibm
set TPCC_SQLLIB=C:\Progra~1\IBM\sqllib
set TPCC_RUNDATA=%HOME%\tpc-c.ibm\tpccdata

@REM TPCC Debug Configuration
@REM This is the path where all error and debug logs are
@REM placed.
@REM To get debugging from within the stored procedures,
@REM you must
@REM set DB2ENVLIST="TPCC_DEBUGDIR" in tpcc.config.
set TPCC_DEBUGDIR=c:\temp

@REM Specifies where stored procedures should be placed
@REM and if they should
@REM be fenced.
set TPCC_SPDIR=%TPCC_SQLLIB%\function
set TPCC_FENCED=NO

```

A.2 Client Transaction Code

Makefile.config

```

#####
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp.
## 1996 - 2006
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication
## or
## disclosure restricted by GSA ADP Schedule Contract with
## IBM Corp.
#####
#####
#
# Makefile.config - AIX 64-bit
#

# Make Configuration
MAKE=make

# Compiler Configuration.
# CFLAGS_DEBUG may be set to "-g", "-DDEBUGIT" "-g -
DDEBUGIT" or left blank
CC=xlC

```

```

CFLAGS_OS=-qflag=i -qlanglvl=ansi -qpluscmt -DSQLUNIX -
DSQLAIX -q64 -O3 -D_LARGE_FILES
CFLAGS_OUT=-o
CFLAGS_DEBUG=

```

```

# Linker Configuration
LD_EXEC=xlC
LD_STORP=xlC
LD_FLAGS_EXEC=-q64
LD_FLAGS_SHLIB=-qmkshrobj
LD_FLAGS_STORP=$(LD_FLAGS_SHLIB) -b:$@.exp -lc -b64
LD_FLAGS_LIB=-L$(TPCC_SQLLIB)/lib -ldb2
LD_FLAGS_OUT=-o

```

```

# Library Configuration
AR=ar
AR_FLAGS=-r -v -X64
AR_FLAGS_LIB=
AR_FLAGS_OUT=

```

```

# OS Commands
ERASE=rm -f
ERASEDIR=$(ERASE) -R
MOVE=mv
COPY=cp

```

```

# OS File Extensions & Path Separators
OBJEXT=.o
LIBEXT=.a
SHLIBEXT=.a
BINEXT=
SLASH=/
CMDSEP=;

```

Src.Common/Makefile

```

#####
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp.
## 1996 - 2006
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication
## or
## disclosure restricted by GSA ADP Schedule Contract with
## IBM Corp.
#####
#####
#
# Makefile - Makefile for Src.Common
#

include $(TPCC_ROOT)/Makefile.config

```

```

#
#####
#####
# Preprocessor, Compiler and Linker Flags
#
#####
#####

BND_OPTS = GRANT PUBLIC \
            MESSAGES $.bnd.msg
PRP_OPTS = BINDFILE \
            OPTLEVEL 1 \
            ISOLATION RR \
            MESSAGES $.prep.msg \
            LEVEL $(TPCC_VERSION) \
            NOLINEMACRO

INCLUDE = -I$(TPCC_SQLLIB)/include -
          I$(TPCC_ROOT)/include

CFLAGS = $(CFLAGS_OS) $(CFLAGS_DEBUG) $(INCLUDE) \
         -DSQLA_NOLINES -D$(DB2EDITION) -
         D$(DB2VERSION) \
         -D$(TPCC_SPTYPE)

UTIL_OBJ = tpcmisc$(OBJEXT) tpccdbg$(OBJEXT)
          tpccctx$(OBJEXT)

#
#####
#####
# User Targets
#
#####
#####

all: connect $(UTIL_OBJ) disconnect

clean:
    - $(ERASE) *$(OBJEXT) *.bnd *.msg tpccctx.c

#
#####
#####
# Helper Targets
#
#####
#####

connect:
    - db2 connect to $(TPCC_DBNAME)

disconnect:
    - db2 connect reset
    - db2 terminate

rebind: connect
        db2 bind tpccctx.bnd $(BND_OPTS)

#
#####
#####

```

```

# Build Rules
#
#####
#####

.SUFFIXES:
.SUFFIXES: $(OBJEXT) .c .sqc

.sqc.c:
    @echo "Prepping $.sqc"
    -db2 prep $.sqc $(PRP_OPTS)
    @echo "Binding $.bnd"
    db2 bind $.bnd $(BND_OPTS)

#
#####
#####
# Dependencies
#
#####
#####

# Source
tpccdbg$(OBJEXT): tpccdbg.c
tpccctx$(OBJEXT): tpccctx.c
tpccmisc$(OBJEXT): tpcmisc.c

# Headers
tpccdbg.c: $(TPCC_ROOT)/include/db2tpcc.h

Src.Common/tpccctx.sqc

/******
****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp.
    1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
    IBM Corp.
*****
****/

/*
 * tpccctx.sqc - TPCC context code
 */

#include <stdlib.h>
#include <stdio.h>
#include <sqlutil.h>
#include "db2tpcc.h"

int connect_to_TM(char *in_dbname);
int connect_to_TM_auth(char *in_dbname, char *in_username,
char *in_password);

```

```

int disconnect_from_TM(void);

int connect_to_TM(char *in_dbname)
{
    return connect_to_TM_auth(in_dbname, "", "");
}

int connect_to_TM_auth(char *in_dbname, char *in_username,
char *in_password)
{
    SQL_STRUCTURE sqlca sqlca;
    int ConnectSQLCODE = 0;

    EXEC SQL BEGIN DECLARE SECTION;
    char dbname[9];
    char username[129];
    char password[15];
    EXEC SQL END DECLARE SECTION;

    strncpy(dbname,in_dbname,8);
    if (strcmp(in_username,"") == 0)
    {
        EXEC SQL CONNECT TO :dbname IN SHARE MODE;
    } else {
        strncpy(username,in_username,128);
        strncpy(password,in_password,14);
        EXEC SQL CONNECT TO :dbname IN SHARE MODE
        USER :username USING :password;
    }

    ConnectSQLCODE = SQLCODE;
    if (ConnectSQLCODE != 0)
    {
        sqlerror( CLIENT_SQL, "CONNECT", __FILE__, __LINE__,
        &sqlca);

        return ConnectSQLCODE;
    }

    return 0;
}

int disconnect_from_TM(void)
{
    SQL_STRUCTURE sqlca sqlca;
    int DisconnectSQLCODE = 0;

    EXEC SQL CONNECT RESET;

    DisconnectSQLCODE = SQLCODE;
    if (DisconnectSQLCODE != 0) {
        sqlerror( CLIENT_SQL, "DISCONNECT", __FILE__,
        __LINE__, &sqlca);
    }

    if (DisconnectSQLCODE) {
        return DisconnectSQLCODE;
    }
    return 0;
}

```

Src.Common/tpccdbg.c

```
/*-----*/
****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp.
1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
*****/

/*
 * tccdbg.c - Debugging Routines
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <time.h>

#include "sqlca.h"
#include "sql.h"
#include "db2tpcc.h"
#include "tpccdbg.h"

#define DEBUG_FILENAME_SZ 128
#define DEBUG_PATH_SIZE 128

void del_print();
void new_print();
void ord_print();
void pay_print();
void stk_print();

void current_tmstamp(char *buf);

static int debugInit = 0;
static char debugPath[DEBUG_PATH_SIZE] = "";

/*-----*/
/* InitializeDebug */
/*-----*/
__inline void InitializeDebug(void) {
    if (debugInit == 0) {
        char *p = getenv("TPCC_DEBUGDIR");
        if (p) {
            strncpy(debugPath, p, DEBUG_PATH_SIZE);
        } else {
            strcpy(debugPath, "/tmp");
        }
        strcat(debugPath, "/");
    }
    debugInit = 1;
}
```

```

}

/*-----*/
/* sqlerror */
/*-----*/
void sqlerror(int tranType, char *msg, char *file, int line,
SQL_STRUCTURE sqlca *psqlca)
{
    FILE *err_fp = NULL;
    char err_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];
    char tranName[16];
    int j,k;
    char timeStamp[27];
    char errStr[512] = "";

    InitializeDebug();
    strncpy(err_fn, debugPath, DEBUG_PATH_SIZE);
    current_tmstamp(&timeStamp[0]);
    timeStamp[19] = (char)NULL;

    switch(tranType)
    {
        case NEWORD_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "new.err.out");
            strcpy(tranName, "NEW_ORDER");
            break;

        case DELIVERY_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "del.err.out");
            strcpy(tranName, "DELIVERY");
            break;

        case PAYMENT_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "pay.err.out");
            strcpy(tranName, "PAYMENT");
            break;

        case ORDSTAT_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "ord.err.out");
            strcpy(tranName, "ORDER_STAT");
            break;

        case STOCKLEV_SQL:
            //sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "stk.err.out");
            strcpy(tranName, "STOCK_LVL");
            break;

        case 0:
            strcat(err_fn, "cli.err.out");
            strcpy(tranName, "CLIENT");
            break;

        default:
            return;
    }
}
```

```
/* Generate Formatted Error Message */
sqlaintp(errStr, 512, 78, psqlca);

err_fp = fopen(err_fn, "a+");

fprintf(err_fp, "-----\n");
fprintf(err_fp, "Transaction: %s (%s)\n", tranName, msg);
fprintf(err_fp, "FILE %s (%u)\n", file, line);
fprintf(err_fp, "SQLCODE %d ", psqlca->sqlcode);
fprintf(err_fp, "PID %d ", getpid());
fprintf(err_fp, "TIME %s\n", timeStamp);
fprintf(err_fp, "-----\n");
fprintf(err_fp, "%s", errStr);
fprintf(err_fp, "-----\n");

if (psqlca->sqlerrmc[0] != ' ' || psqlca->sqlerrmc[1] != ' ')
{
    fprintf(err_fp, "slerrmc: ");

    for(j = 0; j < 5; j++)
    {
        for(k = 0; k < 16; k++) {
            int pos = j * 16 + k;
            if (pos < 70) fprintf(err_fp, "%02x ", psqlca-
>sqlerrmc[pos]);
            else fprintf(err_fp, " ");
        }
        fprintf(err_fp, " |");
        for(k = 0; k < 16; k++) {
            int pos = j * 16 + k;
            char c = ' ';
            if (pos < 70) {
                c = psqlca->sqlerrmc[pos];
                if (!isprint(c)) c = ' ';
            }
            fprintf(err_fp, "%c", c);
        }
        fprintf(err_fp, "\n");
        if (j < 4) fprintf(err_fp, " ");
    }
}

fprintf(err_fp, "sqlerrp: ");
for(j = 0; j < 8; j++)
    fprintf(err_fp, "%c", psqlca->sqlerrp[j]);
fprintf(err_fp, "\n");

fprintf(err_fp, "sqlerrd: ");
for(j = 0; j < 6; j++)
    fprintf(err_fp, " %d", psqlca->sqlerrd[j]);
fprintf(err_fp, "\n");

if (psqlca->sqlwarn[0] != ' ')
{
    fprintf(err_fp, "sqlwarn: ");
    for(j = 0; j < 8; j++)
        fprintf(err_fp, "%c ", psqlca->sqlwarn[j]);
    fprintf(err_fp, "\n");
}

fprintf(err_fp, "\n");
```

```

fclose(err_fp);
}

/*-----*/
/* del_debug */
/*-----*/
void del_debug (struct out_delivery_struct *delivery_ptr,
                struct in_delivery_struct *in_delivery,
                char *msg)
{
    char debug_fn[DEBUG_PATH_SIZE +
                DEBUG_FILENAME_SZ];

    InitializeDebug();
    strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
    strcat(debug_fn, "del.debug.out");
    del_print(delivery_ptr, in_delivery, debug_fn, msg);
}

/*-----*/
/* del_print */
/*-----*/
void del_print (struct out_delivery_struct *delivery_ptr,
                struct in_delivery_struct *in_delivery,
                char *filename,
                char *msg)
{
    FILE *debug_fp;
    char timeStamp[27];
    int j;

    current_tmstamp(&timeStamp[0]);
    timeStamp[19] = (char)NULL;

    if ((debug_fp = fopen(filename, "a+")) == NULL)
    {
        return;
    }

    fprintf(debug_fp, "Delivery debug information follows %s
    (%s)\n", timeStamp, msg);
    fprintf(debug_fp, " PID %d ", getpid());

    fprintf(debug_fp, "\n=====
    =====\n");

    fprintf(debug_fp, "in_delivery_struct {\n");
    fprintf(debug_fp, "ts_W_ID = %d (%X)\n",
            in_delivery->s_W_ID, in_delivery->s_W_ID);
    fprintf(debug_fp, "ts_O_CARRIER_ID = %d (%X)\n",
            in_delivery->s_O_CARRIER_ID, in_delivery-
    >s_O_CARRIER_ID);
    fprintf(debug_fp, "ts_O_DELIVERY_D = %lld (%lX)\n",
            in_delivery->s_O_DELIVERY_D_time, in_delivery-
    >s_O_DELIVERY_D_time);
    fprintf(debug_fp, ")\n\n");

    fprintf(debug_fp, "out_delivery_struct {\n");
    fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
            delivery_ptr->s_transtatus, delivery_ptr->s_transtatus);
    fprintf(debug_fp, "tdeadlocks = %d (%X)\n",

```

```

delivery_ptr->deadlocks, delivery_ptr->deadlocks);

for (j = 0; j < 10; j++) {
    fprintf(debug_fp, "\tts_O_ID[%d] = %d\n",
            j, delivery_ptr->s_O_ID[j]);
}
fprintf(debug_fp, "\t)\n\n");
fclose(debug_fp);
}

/*-----*/
/* new_debug */
/*-----*/
void new_debug (struct out_neword_struct *neword_ptr,
                struct in_neword_struct *in_neword,
                char *msg)
{
    char debug_fn[DEBUG_PATH_SIZE +
                DEBUG_FILENAME_SZ];

    InitializeDebug();
    strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
    strcat(debug_fn, "new.debug.out");
    new_print(neword_ptr, in_neword, debug_fn, msg);
}

/*-----*/
/* new_print */
/*-----*/
void new_print (struct out_neword_struct *neword_ptr,
                struct in_neword_struct *in_neword,
                char *filename,
                char *msg)
{
    FILE *debug_fp;
    char timeStamp[27];
    int j, items;

    current_tmstamp(&timeStamp[0]);
    timeStamp[19] = (char)NULL;

    if ((debug_fp = fopen(filename, "a+")) == NULL)
    {
        return;
    }

    fprintf(debug_fp, "New order debug information follows %s
    (%s)\n", timeStamp, msg);
    fprintf(debug_fp, " PID %d ", getpid());

    fprintf(debug_fp, "\n=====
    =====\n");

    fprintf(debug_fp, "in_neword_struct {\n");

    fprintf(debug_fp, "ts_C_ID = %d (%X)\n",
            in_neword->s_C_ID, in_neword->s_C_ID);
    fprintf(debug_fp, "ts_W_ID = %d (%X)\n",
            in_neword->s_W_ID, in_neword->s_W_ID);
    fprintf(debug_fp, "ts_D_ID = %d (%X)\n",
            in_neword->s_D_ID, in_neword->s_D_ID);

```

```

fprintf(debug_fp, "ts_O_OL_CNT = %d (%X)\n",
        in_neword->s_O_OL_CNT, in_neword->s_O_OL_CNT);
fprintf(debug_fp, "ts_all_local = %d (%X)\n",
        in_neword->s_all_local, in_neword->s_all_local);
fprintf(debug_fp, "ts_O_ENTRY_D = %lld (%lX)\n",
        in_neword->s_O_ENTRY_D_time, in_neword-
>s_O_ENTRY_D_time);
// fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
// in_neword->s_transtatus, in_neword->s_transtatus);
// fprintf(debug_fp, "tduplicate_items= %d (%X)\n",
// in_neword->duplicate_items, in_neword-
>duplicate_items);

fprintf(debug_fp, "titems {\n");
items = in_neword->s_O_OL_CNT;
for (j=0; j<items; j++) {
    if (j != 0)
        fprintf(debug_fp, "\n");
    fprintf(debug_fp, "\tts_OL_I_ID[%d] = %d (%X)\n",
            j, in_neword->in_item[j].s_OL_I_ID, in_neword-
>in_item[j].s_OL_I_ID);
    fprintf(debug_fp, "\tts_OL_SUPPLY_W_ID[%d] = %d
    (%X)\n",
            j, in_neword->in_item[j].s_OL_SUPPLY_W_ID,
            in_neword->in_item[j].s_OL_SUPPLY_W_ID);
    fprintf(debug_fp, "\tts_OL_QUANTITY[%d] = %d (%X)\n",
            j, in_neword->in_item[j].s_OL_QUANTITY, in_neword-
>in_item[j].s_OL_QUANTITY);
}
fprintf(debug_fp, "\t)\n\n");

fprintf(debug_fp, "out_neword_struct {\n");
fprintf(debug_fp, "ts_C_LAST = %s\n",
        neword_ptr->s_C_LAST);
fprintf(debug_fp, "ts_C_CREDIT = %s\n",
        neword_ptr->s_C_CREDIT);
fprintf(debug_fp, "ts_W_TAX = %d\n",
        neword_ptr->s_W_TAX);
fprintf(debug_fp, "ts_D_TAX = %d\n",
        neword_ptr->s_D_TAX);
fprintf(debug_fp, "ts_C_DISCOUNT = %d\n",
        neword_ptr->s_C_DISCOUNT);
fprintf(debug_fp, "ts_O_ID = %d (%X)\n",
        neword_ptr->s_O_ID, neword_ptr->s_O_ID);
fprintf(debug_fp, "ts_O_OL_CNT = %d (%X)\n",
        neword_ptr->s_O_OL_CNT, neword_ptr-
>s_O_OL_CNT);
fprintf(debug_fp, "ts_O_ENTRY_D = %lld (%lX)\n",
        neword_ptr->s_O_ENTRY_D_time, neword_ptr-
>s_O_ENTRY_D_time);
fprintf(debug_fp, "ts_total_amount = %d\n",
        neword_ptr->s_total_amount);
fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
        neword_ptr->s_transtatus, neword_ptr->s_transtatus);
fprintf(debug_fp, "tdeadlocks = %d (%X)\n",
        neword_ptr->deadlocks, neword_ptr->deadlocks);

// fprintf(debug_fp, "ts_W_ID = %d (%X)\n",
// neword_ptr->s_W_ID, neword_ptr->s_W_ID);
// fprintf(debug_fp, "ts_D_ID = %d (%X)\n",
// neword_ptr->s_D_ID, neword_ptr->s_D_ID);
// fprintf(debug_fp, "ts_all_local = %d (%X)\n",

```



```

fprintf(debug_fp,"ts_D_ID = %d (%X)\n",
        in_payment->s_D_ID, in_payment->s_D_ID);
fprintf(debug_fp,"ts_C_D_ID = %d (%X)\n",
        in_payment->s_C_D_ID, in_payment->s_C_D_ID);
fprintf(debug_fp,"ts_C_W_ID = %d (%X)\n",
        in_payment->s_C_W_ID, in_payment->s_C_W_ID);
fprintf(debug_fp,"ts_C_LAST = %s\n",
        in_payment->s_C_LAST);
fprintf(debug_fp,"ts_H_DATE = %lld (%lX)\n",
        in_payment->s_H_DATE_time, in_payment-
>s_H_DATE_time);
fprintf(debug_fp,"\n\n");

fprintf(debug_fp,"out_payment_struct {\n");
fprintf(debug_fp,"ts_H_DATE = %lld (%lX)\n",
        in_payment->s_H_DATE_time, in_payment-
>s_H_DATE_time);
fprintf(debug_fp,"ts_C_CREDIT_LIM = %lld\n",
        payment_ptr->s_C_CREDIT_LIM);
fprintf(debug_fp,"ts_C_DISCOUNT = %d\n",
        payment_ptr->s_C_DISCOUNT);
fprintf(debug_fp,"ts_C_BALANCE = %lld\n",
        payment_ptr->s_C_BALANCE);
fprintf(debug_fp,"ts_C_ID = %d (%X)\n",
        payment_ptr->s_C_ID, payment_ptr->s_C_ID);
fprintf(debug_fp,"ts_W_STREET_1 = %s\n",
        payment_ptr->s_W_STREET_1);
fprintf(debug_fp,"ts_W_STREET_2 = %s\n",
        payment_ptr->s_W_STREET_2);
fprintf(debug_fp,"ts_W_CITY = %s\n",
        payment_ptr->s_W_CITY);
fprintf(debug_fp,"ts_W_STATE = %s\n",
        payment_ptr->s_W_STATE);
fprintf(debug_fp,"ts_W_ZIP = %s\n",
        payment_ptr->s_W_ZIP);
fprintf(debug_fp,"ts_D_STREET_1 = %s\n",
        payment_ptr->s_D_STREET_1);
fprintf(debug_fp,"ts_D_STREET_2 = %s\n",
        payment_ptr->s_D_STREET_2);
fprintf(debug_fp,"ts_D_CITY = %s\n",
        payment_ptr->s_D_CITY);
fprintf(debug_fp,"ts_D_STATE = %s\n",
        payment_ptr->s_D_STATE);
fprintf(debug_fp,"ts_D_ZIP = %s\n",
        payment_ptr->s_D_ZIP);
fprintf(debug_fp,"ts_C_FIRST = %s\n",
        payment_ptr->s_C_FIRST);
fprintf(debug_fp,"ts_C_MIDDLE = %s\n",
        payment_ptr->s_C_MIDDLE);
fprintf(debug_fp,"ts_C_LAST = %s\n",
        payment_ptr->s_C_LAST);
fprintf(debug_fp,"ts_C_STREET_1 = %s\n",
        payment_ptr->s_C_STREET_1);
fprintf(debug_fp,"ts_C_STREET_2 = %s\n",
        payment_ptr->s_C_STREET_2);
fprintf(debug_fp,"ts_C_CITY = %s\n",
        payment_ptr->s_C_CITY);
fprintf(debug_fp,"ts_C_STATE = %s\n",
        payment_ptr->s_C_STATE);
fprintf(debug_fp,"ts_C_ZIP = %s\n",
        payment_ptr->s_C_ZIP);
fprintf(debug_fp,"ts_C_PHONE = %s\n",

```

```

        payment_ptr->s_C_PHONE);
fprintf(debug_fp,"ts_C_SINCE = %lld (%lX)\n",
        payment_ptr->s_C_SINCE_time, payment_ptr-
>s_C_SINCE_time);
fprintf(debug_fp,"ts_C_CREDIT = %s\n",
        payment_ptr->s_C_CREDIT);
fprintf(debug_fp,"ts_C_DATA = %s\n",
        payment_ptr->s_C_DATA);
fprintf(debug_fp,"ts_transtatus = %d (%X)\n",
        payment_ptr->s_transtatus, payment_ptr->s_transtatus);
fprintf(debug_fp,"tdeadlocks = %d (%X)\n",
        payment_ptr->deadlocks, payment_ptr->deadlocks);
fprintf(debug_fp,"\n\n");
fclose(debug_fp);
}

/*-----*/
/* stk_debug */
/*-----*/
void stk_debug (struct out_stocklev_struct *stocklev,
                struct in_stocklev_struct *in_stocklev,
                char *msg)
{
    char debug_fn[DEBUG_PATH_SIZE +
DEBUG_FILENAME_SZ];

    InitializeDebug();
    strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
    strcat(debug_fn, "stk.debug.out");
    stk_print(stocklev, in_stocklev, debug_fn, msg);
}

/*-----*/
/* stk_print */
/*-----*/
void stk_print (struct out_stocklev_struct *stocklev,
                struct in_stocklev_struct *in_stocklev,
                char *filename,
                char *msg)
{
    FILE *debug_fp;
    char timeStamp[27];

    current_tmstamp(&timeStamp[0]);
    timeStamp[19] = (char)NULL;

    if ((debug_fp = fopen(filename, "a+")) == NULL)
    {
        return;
    }

    fprintf(debug_fp,"Stock level debug information follows %s
(%s)\n", timeStamp, msg);
    fprintf(debug_fp, " PID %d ", getpid());

    fprintf(debug_fp, "\n=====
=====");

    fprintf(debug_fp, "in_stocklev_struct {\n");
    fprintf(debug_fp,"ts_W_ID = %d (%X)\n",
        in_stocklev->s_W_ID, in_stocklev->s_W_ID);

```

```

fprintf(debug_fp,"ts_D_ID = %d (%X)\n",
        in_stocklev->s_D_ID, in_stocklev->s_D_ID);
fprintf(debug_fp,"ts_threshold = %d (%X)\n",
        in_stocklev->s_threshold, in_stocklev->s_threshold);
fprintf(debug_fp,")\n\n");

fprintf(debug_fp,"out_stocklev_struct {\n");
fprintf(debug_fp,"ts_transtatus = %d (%X)\n",
        stocklev->s_transtatus, stocklev->s_transtatus);
fprintf(debug_fp,"tdeadlocks = %d (%X)\n",
        stocklev->deadlocks, stocklev->deadlocks);
fprintf(debug_fp,"ts_low_stock = %d (%X)\n",
        stocklev->s_low_stock, stocklev->s_low_stock);
fprintf(debug_fp,")\n\n");
fclose(debug_fp);
}

void current_tmstamp(char *buf)
{
    time_t t = time(NULL);
    strncpy(buf,ctime(&t),19);
}

Src.Common/tpccmisc.c

/*****
** Licensed Materials - Property of IBM
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
** (C) COPYRIGHT International Business Machines Corp.
1996 - 2006
** All Rights Reserved.
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
****/

/*
 * tpccmisc.c - Miscellaneous routines
 */

#include <stdlib.h>
#include <sys/types.h>
#include <sys/time.h>

double current_time_ms(void);
double current_time(void);

/* Current time in SECONDS, precision SECONDS */
double current_time(void)
{
    /* use time() to get seconds */
    return(time(NULL));
}

```

```

/* Current time in SECONDS, precision MILLISECONDS */
double current_time_ms(void)
{
    /* gettimeofday() returns seconds and microseconds */
    /* convert to fractional seconds */
    struct timeval t;
    gettimeofday(&t,NULL);
    return (t.tv_sec + (double)t.tv_usec/(1000*1000));
}

```

Src.Srv/Makefile

```

#####
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp.
## 1996 - 2006
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication
## or
## disclosure restricted by GSA ADP Schedule Contract with
## IBM Corp.
#####
#####
#
# Makefile - Makefile for Src.Srv
#

include $(TPCC_ROOT)/Makefile.config

#
#####
#####
# Preprocessor, Compiler and Linker Flags
#
#####
#####

BND_OPTS = GRANT PUBLIC \
            MESSAGES $.bnd.msg
PRP_OPTS = BINDFILE \
            EXPLAIN ALL \
            MESSAGES $.prep.msg

INCLUDE = -I$(TPCC_SQLLIB)/include -
I$(TPCC_ROOT)/include

CFLAGS = $(CFLAGS_OS) $(INCLUDE) $(CFLAGS_DEBUG) \
         -D$(DB2EDITION) -D$(DB2VERSION) \
         -DSQLA_NOLINES -DLINT_ARGS

LDFLAGS = $(LDFLAGS_STORP) $(LDFLAGS_LIB)

```

```

#
#####
#####
# File Collections
#
#####
#####

STORED_PROCS = new ord del

UTIL_OBJ =
            $(TPCC_ROOT)/Src.Common/tpccmisc$(OBJEXT) \
            $(TPCC_ROOT)/Src.Common/tpccdbg$(OBJEXT)

EXE = news ords dels

#
#####
#####
# User Targets
#
#####
#####

all: connect explain catalog $(EXE) install plan disconnect

clean: connect uncatlog unexplain disconnect
      - $(ERASE) $(TPCC_SPDIR)$(SLASH)news
      - $(ERASE) $(TPCC_SPDIR)$(SLASH)ords
      - $(ERASE) $(TPCC_SPDIR)$(SLASH)dels
      - $(ERASE) *.bnd *.msg *.out *$(OBJEXT) $(EXE)
tpcc_all_sql.c
  - $(ERASE) TPCC_ALL.*.plan

#
#####
#####
# Helper Targets
#
#####
#####

catalog: uncatlog
         - perl $(TPCC_ROOT)$(SLASH)utils$(SLASH)genproc.pl
$(STORED_PROCS)
         - db2 -tvf cat-proc.ddl +o -z cat-proc.out
         - db2 -td$$$ -vf cat-func.ddl +o -z cat-func.out

uncatlog:
         - perl $(TPCC_ROOT)$(SLASH)utils$(SLASH)genproc.pl
$(STORED_PROCS)
         - db2 -td$$$ -vf uncat-func.ddl +o -z uncat-func.out
         - db2 -tvf uncat-proc.ddl +o -z uncat-proc.out

explain:
         - perl
$(TPCC_ROOT)$(SLASH)utils$(SLASH)fixup_explain.pl
         - db2 -tvf
$(TPCC_ROOT)$(SLASH)utils$(SLASH)EXPLAIN.DDL +o -z
EXPLAIN.out

unexplain:

```

```

         - db2 -tvf
$(TPCC_ROOT)$(SLASH)utils$(SLASH)UNEXPLAIN.DDL +o -z
UNEXPLAIN.out

connect:
         - db2 connect to $(TPCC_DBNAME)

disconnect:
         - db2 connect reset
         - db2 terminate

plan:
         - db2exfmt -d $(TPCC_DBNAME) -e $(TPCC_SCHEMA) -
s $(TPCC_SCHEMA) -w -1 -n TPCC_ALL -g -# 0 -o
TPCC_ALL.exfmt.plan
         - (export DB2EXPLN_BUFFER=3000000; db2expln -d
$(TPCC_DBNAME) -c $(TPCC_SCHEMA) -p TPCC_ALL -s 0 -
g -o TPCC_ALL.expln.plan )

rebind: connect catalog
         db2 bind tpcc_all_sql.bnd $(BND_OPTS) QUERYOPT 7

#
#####
#####
# Install Targets
#
#####
#####

install: $(EXE)
         - mkdir $(TPCC_SPDIR)
$(COPY) ords $(TPCC_SPDIR)
$(COPY) news $(TPCC_SPDIR)
$(COPY) dels $(TPCC_SPDIR)

#
#####
#####
# Build Rules
#
#####
#####

.SUFFIXES: $(OBJEXT) .c .sqc

# d230437mte: QUERYOPT 7 required for UNION ALL
# Only stock needs CS , and that can be specified on the
SELECT statement
tpcc_all_sql.c:
    @echo "Prepping $.sqc"
    -db2 prep $.sqc $(PRP_OPTS) ISOLATION RR
    @echo "Binding $.bnd"
    db2 bind $.bnd $(BND_OPTS) QUERYOPT 7

# Stored procedures are built in a special way

tpcc_all_sql$(OBJEXT):
    $(CC) -c tpcc_all_sql.c $(CFLAGS) -D$(TPCC_SPTYPE)
$(CFLAGS_OUT)$@

$(EXE): $(UTIL_OBJ) tpcc_all_sql.o

```

```

$(LD_STORP) $(LD_FLAGS) $(UTIL_OBJ) tpcc_all_sql.o
$(LD_FLAGS_OUT)$@

#
#####
#####
# Dependencies
#
#####
#####

# Executables (Stored Procedures)
$(EXE): $(UTIL_OBJ) tpcc_all_sql.o

# Source
tpcc_all_sql$(OBJEXT): tpcc_all_sql.c

# Headers
tpcc_all_sql.c: $(TPCC_ROOT)/include/db2tpcc.h

```

Src.Srv/cat-func.ddl

```

-----
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996
-- 2006
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with
-- IBM Corp.
-----
--
-- cat-func.ddl - Create table functions
--
--
-- DELIVERY
--
CREATE FUNCTION DEL( W_ID      INTEGER
                   , D_ID      SMALLINT
                   , CARRIER_ID SMALLINT
                   , DELIVERY_D BIGINT
                   )
RETURN TABLE ( O_ID INTEGER )
SPECIFIC DELIVERY
MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL
ACTION LANGUAGE SQL
VAR: BEGIN ATOMIC
DECLARE O_ID INTEGER ;

```

```

DECLARE C_ID INTEGER ;
DECLARE AMOUNT INTEGER ;

/* Delete the order from new order table */
SET VAR.O_ID = ( SELECT NO_O_ID
                FROM OLD TABLE ( DELETE
                                FROM ( SELECT NO_O_ID
                                        FROM NEW_ORDER
                                        WHERE NO_W_ID = DEL.W_ID
                                        AND NO_D_ID = DEL.D_ID
                                        ORDER BY NO_O_ID ASC
                                        FETCH FIRST 1 ROW ONLY
                                    ) AS NEW_ORDER
                                ) AS D
                ) ;

/* Update the order as delivered and retrieve the customer id */
SET VAR.C_ID = ( SELECT O_C_ID
                FROM OLD TABLE ( UPDATE ORDERS
                                SET O_CARRIER_ID =
DEL.CARRIER_ID
                                WHERE O_W_ID = DEL.W_ID
                                AND O_D_ID = DEL.D_ID
                                AND O_ID = VAR.O_ID
                                ) AS U
                ) ;

SET VAR.AMOUNT = ( SELECT SUM( OL_AMOUNT )
                  FROM OLD TABLE ( UPDATE ORDER_LINE
                                SET OL_DELIVERY_D =
DEL.DELIVERY_D
                                WHERE OL_W_ID = DEL.W_ID
                                AND OL_D_ID = DEL.D_ID
                                AND OL_O_ID = VAR.O_ID
                                ) AS U
                  ) ;

/* Charge the customer */
UPDATE CUSTOMER
SET C_BALANCE = C_BALANCE + VAR.AMOUNT
   , C_DELIVERY_CNT = C_DELIVERY_CNT +
SMALLINT( 1 )

```

```

WHERE C_W_ID = DEL.W_ID
      AND C_D_ID = DEL.D_ID
      AND C_ID = VAR.C_ID
;

/* Return the order id to the caller (or NULL) */
RETURN VALUES VAR.O_ID ;

END
$

--
-- ORDER STATUS
--

CREATE FUNCTION ORD_C_LAST( W_ID INTEGER
                           , D_ID SMALLINT
                           , C_LAST VARCHAR(16)
                           )
RETURNS TABLE( O_ID      INTEGER
               , O_CARRIER_ID SMALLINT
               , O_ENTRY_D  BIGINT
               , C_BALANCE  BIGINT
               , C_FIRST    VARCHAR(16)
               , C_MIDDLE   CHAR(2)
               , C_ID       INTEGER
               )
SPECIFIC ORD_C_LAST
READS SQL DATA NO EXTERNAL ACTION DETERMINISTIC
LANGUAGE SQL
VAR: BEGIN ATOMIC

DECLARE C_BALANCE  BIGINT ;
DECLARE C_FIRST    VARCHAR(16) ;
DECLARE C_MIDDLE   CHAR(2) ;
DECLARE C_ID       INTEGER ;
DECLARE O_ID       INTEGER ;
DECLARE O_CARRIER_ID SMALLINT ;
DECLARE O_ENTRY_D  BIGINT ;

/* Retrieve the Customer information */
SET ( C_BALANCE, C_FIRST, C_MIDDLE, C_ID )
= ( SELECT C_BALANCE, C_FIRST, C_MIDDLE , C_ID
    FROM ( SELECT C_ID
           , C_BALANCE
           , C_FIRST
           , C_MIDDLE
           , COUNT(*) OVER() AS COUNT
           , ROWNUMBER() OVER (ORDER BY C_FIRST)
    AS NUM
    FROM CUSTOMER
    WHERE C_W_ID = ORD_C_LAST.W_ID

```

```

        AND C_D_ID = ORD_C_LAST.D_ID
        AND C_LAST = ORD_C_LAST.C_LAST

    ) AS V1

    WHERE NUM = (COUNT + BIGINT( 1 )) / BIGINT( 2 )
    )
;

/* Take advantage of the index to fetch the first row (and hence
max(o_id) ) */

SET ( O_ID , O_CARRIER_ID , O_ENTRY_D )
= ( SELECT O_ID
    , O_CARRIER_ID
    , O_ENTRY_D

    FROM ORDERS

    WHERE O_W_ID = ORD_C_LAST.W_ID
    AND O_D_ID = ORD_C_LAST.D_ID
    AND O_C_ID = VAR.C_ID

    ORDER BY O_ID DESC
    FETCH FIRST 1 ROW ONLY

    )
;

RETURN VALUES ( VAR.O_ID
    , VAR.O_CARRIER_ID
    , VAR.O_ENTRY_D
    , VAR.C_BALANCE
    , VAR.C_FIRST
    , VAR.C_MIDDLE
    , VAR.C_ID
    )
;

END $

CREATE FUNCTION ORD_C_ID( W_ID INTEGER
    , D_ID SMALLINT
    , C_ID INTEGER
    )

RETURNS TABLE( O_ID INTEGER
    , O_CARRIER_ID SMALLINT
    , O_ENTRY_D BIGINT
    , C_BALANCE BIGINT
    , C_FIRST VARCHAR(16)
    , C_MIDDLE CHAR(2)
    , C_LAST VARCHAR(16)
    )

SPECIFIC ORD_C_ID

READS SQL DATA NO EXTERNAL ACTION DETERMINISTIC
LANGUAGE SQL

```

```

VAR: BEGIN ATOMIC

DECLARE C_BALANCE BIGINT ;
DECLARE C_FIRST VARCHAR(16) ;
DECLARE C_MIDDLE CHAR(2) ;
DECLARE C_LAST VARCHAR(16) ;
DECLARE O_ID INTEGER ;
DECLARE O_CARRIER_ID SMALLINT ;
DECLARE O_ENTRY_D BIGINT ;

/* Retrieve the Customer information */

SET ( C_BALANCE , C_FIRST , C_MIDDLE , C_LAST )
= ( SELECT C_BALANCE , C_FIRST , C_MIDDLE , C_LAST

    FROM CUSTOMER

    WHERE C_ID = ORD_C_ID.C_ID
    AND C_W_ID = ORD_C_ID.W_ID
    AND C_D_ID = ORD_C_ID.D_ID
    )
;

SET ( O_ID , O_CARRIER_ID , O_ENTRY_D )
= ( SELECT O_ID
    , O_CARRIER_ID
    , O_ENTRY_D

    FROM ORDERS

    WHERE O_W_ID = ORD_C_ID.W_ID
    AND O_D_ID = ORD_C_ID.D_ID
    AND O_C_ID = ORD_C_ID.C_ID

    ORDER BY O_ID DESC
    FETCH FIRST 1 ROW ONLY

    )
;

RETURN VALUES ( VAR.O_ID
    , VAR.O_CARRIER_ID
    , VAR.O_ENTRY_D
    , VAR.C_BALANCE
    , VAR.C_FIRST
    , VAR.C_MIDDLE
    , VAR.C_LAST
    )
;

END $

--
-- PAYMENT
--

CREATE FUNCTION PAY_C_LAST( W_ID INTEGER
    , D_ID SMALLINT
    , C_W_ID INTEGER
    , C_D_ID SMALLINT

```

```

    , C_LAST VARCHAR(16)
    , H_DATE BIGINT
    , H_AMOUNT BIGINT
    , BAD_CREDIT_PREFIX VARCHAR(28)
    )

RETURNS TABLE( W_STREET_1 CHAR(20)
    , W_STREET_2 CHAR(20)
    , W_CITY CHAR(20)
    , W_STATE CHAR(2)
    , W_ZIP CHAR(9)
    , D_STREET_1 CHAR(20)
    , D_STREET_2 CHAR(20)
    , D_CITY CHAR(20)
    , D_STATE CHAR(2)
    , D_ZIP CHAR(9)
    , C_ID INTEGER
    , C_FIRST VARCHAR(16)
    , C_MIDDLE CHAR(2)
    , C_STREET_1 VARCHAR(20)
    , C_STREET_2 VARCHAR(20)
    , C_CITY VARCHAR(20)
    , C_STATE CHAR(2)
    , C_ZIP CHAR(9)
    , C_PHONE CHAR(16)
    , C_SINCE BIGINT
    , C_CREDIT CHAR(2)
    , C_CREDIT_LIM BIGINT
    , C_DISCOUNT INTEGER
    , C_BALANCE BIGINT
    , C_DATA CHAR(200)
    )

SPECIFIC PAY_C_LAST

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL
ACTION LANGUAGE SQL

VAR: BEGIN ATOMIC

DECLARE W_NAME CHAR(10) ;
DECLARE D_NAME CHAR(10) ;

DECLARE W_STREET_1 CHAR(20) ;
DECLARE W_STREET_2 CHAR(20) ;
DECLARE W_CITY CHAR(20) ;
DECLARE W_STATE CHAR(2) ;
DECLARE W_ZIP CHAR(9) ;

DECLARE D_STREET_1 CHAR(20) ;
DECLARE D_STREET_2 CHAR(20) ;
DECLARE D_CITY CHAR(20) ;
DECLARE D_STATE CHAR(2) ;
DECLARE D_ZIP CHAR(9) ;

DECLARE C_ID INTEGER ;

DECLARE C_FIRST VARCHAR(16) ;
DECLARE C_MIDDLE CHAR(2) ;
DECLARE C_STREET_1 VARCHAR(20) ;
DECLARE C_STREET_2 VARCHAR(20) ;
DECLARE C_CITY VARCHAR(20) ;

```

```

DECLARE C_STATE CHAR(2);
DECLARE C_ZIP CHAR(9);
DECLARE C_PHONE CHAR(16);
DECLARE C_SINCE BIGINT;
DECLARE C_CREDIT CHAR(2);
DECLARE C_CREDIT_LIM BIGINT;
DECLARE C_DISCOUNT INTEGER;
DECLARE C_BALANCE BIGINT;
DECLARE C_DATA CHAR(200);

/* Update District and retrieve its data */

SET ( D_NAME, D_STREET_1, D_STREET_2, D_CITY,
D_STATE, D_ZIP)

= ( SELECT D_NAME, D_STREET_1, D_STREET_2,
D_CITY, D_STATE, D_ZIP

FROM OLD TABLE ( UPDATE DISTRICT

SET D_YTD = D_YTD +
PAY_C_LAST.H_AMOUNT

WHERE D_W_ID = PAY_C_LAST.W_ID
AND D_ID = PAY_C_LAST.D_ID
) AS U
);

/* Determine the C_ID */

SET ( C_ID )

= ( SELECT C_ID
FROM ( SELECT C_ID
, COUNT(*) OVER() AS COUNT
, ROWNUMBER() OVER (ORDER BY C_FIRST)
AS NUM

FROM CUSTOMER
WHERE C_LAST = PAY_C_LAST.C_LAST
AND C_W_ID = PAY_C_LAST.C_W_ID
AND C_D_ID = PAY_C_LAST.C_D_ID
) AS T

WHERE NUM = (COUNT + BIGINT( 1 )) / BIGINT( 2 )
);

/* Update the middle customer */

SET ( C_ID, C_FIRST, C_MIDDLE, C_STREET_1,
C_STREET_2
, C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE,
C_CREDIT, C_CREDIT_LIM
, C_DISCOUNT, C_BALANCE, C_DATA )

= ( SELECT C_ID, C_FIRST, C_MIDDLE, C_STREET_1,
C_STREET_2
, C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE,
C_CREDIT, C_CREDIT_LIM

```

```

, C_DISCOUNT, C_BALANCE
, CASE WHEN C_CREDIT = 'BC' THEN
SUBSTR(C_DATA, 1, 200) ELSE NULL END AS C_DATA

FROM NEW TABLE ( UPDATE CUSTOMER

SET C_BALANCE = C_BALANCE -
PAY_C_LAST.H_AMOUNT
, C_YTD_PAYMENT =
C_YTD_PAYMENT + PAY_C_LAST.H_AMOUNT
, C_PAYMENT_CNT =
C_PAYMENT_CNT + SMALLINT( 1 )

, C_DATA = CASE WHEN C_CREDIT =
'BC'
THEN CHAR( C_ID ) --
11 bytes long
|| BAD_CREDIT_PREFIX
-- 28 bytes long
|| SUBSTR( C_DATA, 1,
461 ) -- 461 + 39 = 500
ELSE C_DATA
END

WHERE C_W_ID = PAY_C_LAST.C_W_ID
AND C_D_ID = PAY_C_LAST.C_D_ID
AND C_ID = VAR.C_ID
) AS U
);

/* Update the warehouse */

SET ( W_NAME, W_STREET_1, W_STREET_2, W_CITY,
W_STATE, W_ZIP )

= ( SELECT W_NAME, W_STREET_1, W_STREET_2,
W_CITY, W_STATE, W_ZIP

FROM OLD TABLE ( UPDATE WAREHOUSE

SET W_YTD = W_YTD +
PAY_C_LAST.H_AMOUNT

WHERE W_ID = PAY_C_LAST.W_ID
) AS U
);

/* Finally insert into the warehouse */

INSERT

INTO HISTORY ( H_C_ID, H_C_D_ID, H_C_W_ID, H_D_ID,
H_W_ID, H_DATA, H_DATE, H_AMOUNT )

VALUES ( VAR.C_ID
, PAY_C_LAST.C_D_ID
, PAY_C_LAST.C_W_ID
, PAY_C_LAST.D_ID
, PAY_C_LAST.W_ID
, VAR.W_NAME || CHAR( ' ', 4 ) || VAR.D_NAME

```

```

, PAY_C_LAST.H_DATE
, PAY_C_LAST.H_AMOUNT
);

/* Done - return the collected data */

RETURN VALUES ( W_STREET_1, W_STREET_2,
W_CITY, W_STATE, W_ZIP
, D_STREET_1, D_STREET_2, D_CITY, D_STATE,
D_ZIP
, C_ID, C_FIRST, C_MIDDLE, C_STREET_1,
C_STREET_2
, C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE,
C_CREDIT, C_CREDIT_LIM
, C_DISCOUNT, C_BALANCE, C_DATA
);

END
$

CREATE FUNCTION PAY_C_ID( W_ID INTEGER
, D_ID SMALLINT
, C_W_ID INTEGER
, C_D_ID SMALLINT
, C_ID INTEGER
, H_DATE BIGINT
, H_AMOUNT BIGINT
, BAD_CREDIT_PREFIX VARCHAR(34)
)

RETURNS TABLE( W_STREET_1 CHAR(20)
, W_STREET_2 CHAR(20)
, W_CITY CHAR(20)
, W_STATE CHAR(2)
, W_ZIP CHAR(9)
, D_STREET_1 CHAR(20)
, D_STREET_2 CHAR(20)
, D_CITY CHAR(20)
, D_STATE CHAR(2)
, D_ZIP CHAR(9)
, C_LAST VARCHAR(16)
, C_FIRST VARCHAR(16)
, C_MIDDLE CHAR(2)
, C_STREET_1 VARCHAR(20)
, C_STREET_2 VARCHAR(20)
, C_CITY VARCHAR(20)
, C_STATE CHAR(2)
, C_ZIP CHAR(9)
, C_PHONE CHAR(16)
, C_SINCE BIGINT
, C_CREDIT CHAR(2)
, C_CREDIT_LIM BIGINT
, C_DISCOUNT INTEGER
, C_BALANCE BIGINT
, C_DATA CHAR(200)
)

SPECIFIC PAY_C_ID

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL
ACTION LANGUAGE SQL

```

```

VAR: BEGIN ATOMIC

DECLARE W_NAME CHAR(10);
DECLARE D_NAME CHAR(10);

DECLARE W_STREET_1 CHAR(20);
DECLARE W_STREET_2 CHAR(20);
DECLARE W_CITY CHAR(20);
DECLARE W_STATE CHAR(2);
DECLARE W_ZIP CHAR(9);

DECLARE D_STREET_1 CHAR(20);
DECLARE D_STREET_2 CHAR(20);
DECLARE D_CITY CHAR(20);
DECLARE D_STATE CHAR(2);
DECLARE D_ZIP CHAR(9);

DECLARE C_LAST VARCHAR(16);

DECLARE C_FIRST VARCHAR(16);
DECLARE C_MIDDLE CHAR(2);
DECLARE C_STREET_1 VARCHAR(20);
DECLARE C_STREET_2 VARCHAR(20);
DECLARE C_CITY VARCHAR(20);
DECLARE C_STATE CHAR(2);
DECLARE C_ZIP CHAR(9);
DECLARE C_PHONE CHAR(16);
DECLARE C_SINCE BIGINT;
DECLARE C_CREDIT CHAR(2);
DECLARE C_CREDIT_LIM BIGINT;
DECLARE C_DISCOUNT INTEGER;
DECLARE C_BALANCE BIGINT;
DECLARE C_DATA CHAR(200);

/* Update District and retrieve its data */

SET ( D_NAME, D_STREET_1, D_STREET_2, D_CITY,
D_STATE, D_ZIP )

= ( SELECT D_NAME, D_STREET_1, D_STREET_2,
D_CITY, D_STATE, D_ZIP

FROM OLD TABLE ( UPDATE DISTRICT

SET D_YTD = D_YTD +
PAY_C_ID.H_AMOUNT

WHERE D_W_ID = PAY_C_ID.W_ID
AND D_ID = PAY_C_ID.D_ID
) AS U
)

/* Update the middle customer */

SET ( C_LAST, C_FIRST, C_MIDDLE, C_STREET_1,
C_STREET_2
, C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE,
C_CREDIT, C_CREDIT_LIM
, C_DISCOUNT, C_BALANCE, C_DATA )

```

```

= ( SELECT C_LAST, C_FIRST, C_MIDDLE,
C_STREET_1, C_STREET_2
, C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE,
C_CREDIT, C_CREDIT_LIM
, C_DISCOUNT, C_BALANCE
CASE WHEN C_CREDIT = 'BC' THEN
SUBSTR(C_DATA, 1, 200) ELSE NULL END AS C_DATA

FROM NEW TABLE ( UPDATE CUSTOMER

SET C_BALANCE = C_BALANCE -
PAY_C_ID.H_AMOUNT
, C_YTD_PAYMENT =
C_YTD_PAYMENT + PAY_C_ID.H_AMOUNT
, C_PAYMENT_CNT =
C_PAYMENT_CNT + SMALLINT( 1 )

, C_DATA = CASE WHEN C_CREDIT =
'BC'
THEN BAD_CREDIT_PREFIX
-- 34 bytes long
|| SUBSTR( C_DATA, 1,
466 ) -- 466 + 34 = 500 bytes
ELSE C_DATA
END

WHERE C_W_ID = PAY_C_ID.C_W_ID
AND C_D_ID = PAY_C_ID.C_D_ID
AND C_ID = PAY_C_ID.C_ID
) AS U

/* Update the warehouse */

SET ( W_NAME, W_STREET_1, W_STREET_2, W_CITY,
W_STATE, W_ZIP )

= ( SELECT W_NAME, W_STREET_1, W_STREET_2,
W_CITY, W_STATE, W_ZIP

FROM OLD TABLE ( UPDATE WAREHOUSE

SET W_YTD = W_YTD +
PAY_C_ID.H_AMOUNT

WHERE W_ID = PAY_C_ID.W_ID
) AS U
)

/* Finally insert into the warehouse */

INSERT

INTO HISTORY ( H_C_ID, H_C_D_ID, H_C_W_ID, H_D_ID,
H_W_ID, H_DATA, H_DATE, H_AMOUNT )

VALUES ( PAY_C_ID.C_ID
, PAY_C_ID.C_D_ID
, PAY_C_ID.C_W_ID

```

```

, PAY_C_ID.D_ID
, PAY_C_ID.W_ID
, VAR.W_NAME || CHAR(' ', 4) || VAR.D_NAME
, PAY_C_ID.H_DATE
, PAY_C_ID.H_AMOUNT
)
;

/* Done - return the collected data */

RETURN VALUES ( W_STREET_1, W_STREET_2,
W_CITY, W_STATE, W_ZIP
, D_STREET_1, D_STREET_2, D_CITY, D_STATE,
D_ZIP
, C_LAST, C_FIRST, C_MIDDLE, C_STREET_1,
C_STREET_2
, C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE,
C_CREDIT, C_CREDIT_LIM
, C_DISCOUNT, C_BALANCE, C_DATA
)
;

END
$

--
-- NEW ORDER
--

CREATE FUNCTION NEW_OL_ALL( I_ID INT
, I_QTY SMALLINT
, W_ID INT
, SUPP_W_ID INT
, O_ID INT
, D_ID SMALLINT
)

RETURNS TABLE( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, OL_DIST_INFO CHAR(24)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT
)

SPECIFIC NEW_OL_ALL

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL
ACTION LANGUAGE SQL

VAR: BEGIN ATOMIC

DECLARE I_PRICE INTEGER ;
DECLARE I_NAME CHAR(24) ;
DECLARE I_DATA VARCHAR(50) ;
DECLARE OL_DIST_INFO CHAR(24) ;
DECLARE S_DATA VARCHAR(50) ;
DECLARE S_QUANTITY SMALLINT ;

SET ( I_PRICE, I_NAME, I_DATA )

```

```

= ( SELECT
  I_PRICE
  , I_NAME
  , I_DATA
  FROM ITEM
  WHERE ITEM.I_ID = NEW_OL_ALL.I_ID
);
SET ( OL_DIST_INFO , S_DATA , S_QUANTITY )
= ( SELECT OL_DIST_INFO
  , S_DATA
  , S_QUANTITY
  FROM NEW TABLE ( UPDATE STOCK
    INCLUDE ( OL_DIST_INFO CHAR( 24 ) )
    SET S_QUANTITY = CASE WHEN
S_QUANTITY - NEW_OL_ALL.I_QTY >= 10
    THEN S_QUANTITY -
NEW_OL_ALL.I_QTY
    ELSE S_QUANTITY -
NEW_OL_ALL.I_QTY + 91
    END
    , S_ORDER_CNT =
S_ORDER_CNT + SMALLINT( 1 )
    , S_YTD = S_YTD +
NEW_OL_ALL.I_QTY
    , S_REMOTE_CNT = CASE
WHEN NEW_OL_ALL.SUPP_W_ID = NEW_OL_ALL.W_ID
    THEN
S_REMOTE_CNT
    ELSE
S_REMOTE_CNT + SMALLINT( 1 )
    END
    , OL_DIST_INFO = CASE
D_ID WHEN SMALLINT( 1 ) THEN S_DIST_01
SMALLINT( 2 ) THEN S_DIST_02
SMALLINT( 3 ) THEN S_DIST_03
SMALLINT( 4 ) THEN S_DIST_04
SMALLINT( 5 ) THEN S_DIST_05
SMALLINT( 6 ) THEN S_DIST_06
SMALLINT( 7 ) THEN S_DIST_07
SMALLINT( 8 ) THEN S_DIST_08
SMALLINT( 9 ) THEN S_DIST_09
SMALLINT( 10 ) THEN S_DIST_10

```

```

END
WHERE S_I_ID = NEW_OL_ALL.I_ID
AND S_W_ID =
NEW_OL_ALL.SUPP_W_ID
) AS U
);
RETURN VALUES( VAR.I_PRICE
  , VAR.I_NAME
  , VAR.I_DATA
  , VAR.OL_DIST_INFO
  , VAR.S_DATA
  , VAR.S_QUANTITY
);
END
$
CREATE FUNCTION NEW_OL_LOCAL( I_ID INT
  , I_QTY SMALLINT
  , W_ID INT
  , O_ID INT
  , D_ID SMALLINT
)
RETURNS TABLE( I_PRICE INTEGER
  , I_NAME CHAR(24)
  , I_DATA VARCHAR(50)
  , OL_DIST_INFO CHAR(24)
  , S_DATA VARCHAR(50)
  , S_QUANTITY SMALLINT
)
SPECIFIC NEW_OL_LOCAL
MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL
ACTION LANGUAGE SQL
VAR: BEGIN ATOMIC
DECLARE I_PRICE INTEGER ;
DECLARE I_NAME CHAR(24) ;
DECLARE I_DATA VARCHAR(50) ;
DECLARE OL_DIST_INFO CHAR(24) ;
DECLARE S_DATA VARCHAR(50) ;
DECLARE S_QUANTITY SMALLINT ;
SET ( I_PRICE , I_NAME , I_DATA )
= ( SELECT
  I_PRICE
  , I_NAME
  , I_DATA
  FROM ITEM
  WHERE ITEM.I_ID = NEW_OL_LOCAL.I_ID
);

```

```

SET ( OL_DIST_INFO , S_DATA , S_QUANTITY )
= ( SELECT OL_DIST_INFO
  , S_DATA
  , S_QUANTITY
  FROM NEW TABLE ( UPDATE STOCK
    INCLUDE ( OL_DIST_INFO CHAR( 24 ) )
    SET S_QUANTITY = CASE WHEN
S_QUANTITY - NEW_OL_LOCAL.I_QTY >= 10
    THEN S_QUANTITY -
NEW_OL_LOCAL.I_QTY
    ELSE S_QUANTITY -
NEW_OL_LOCAL.I_QTY + 91
    END
    , S_ORDER_CNT =
S_ORDER_CNT + SMALLINT( 1 )
    , S_YTD = S_YTD +
NEW_OL_LOCAL.I_QTY
    , OL_DIST_INFO = CASE
D_ID WHEN SMALLINT( 1 ) THEN S_DIST_01
SMALLINT( 2 ) THEN S_DIST_02
SMALLINT( 3 ) THEN S_DIST_03
SMALLINT( 4 ) THEN S_DIST_04
SMALLINT( 5 ) THEN S_DIST_05
SMALLINT( 6 ) THEN S_DIST_06
SMALLINT( 7 ) THEN S_DIST_07
SMALLINT( 8 ) THEN S_DIST_08
SMALLINT( 9 ) THEN S_DIST_09
SMALLINT( 10 ) THEN S_DIST_10
    END
    WHERE S_I_ID = NEW_OL_LOCAL.I_ID
AND S_W_ID = NEW_OL_LOCAL.W_ID
) AS U
);
RETURN VALUES( VAR.I_PRICE
  , VAR.I_NAME
  , VAR.I_DATA
  , VAR.OL_DIST_INFO
  , VAR.S_DATA
  , VAR.S_QUANTITY
);
END
$

```

```

CREATE FUNCTION NEW_WH ( O_ID    INTEGER
                      , W_ID    INTEGER
                      , D_ID    SMALLINT
                      , C_ID    INTEGER
                      , O_ENTRY_D BIGINT
                      , O_OL_CNT SMALLINT
                      , O_ALL_LOCAL SMALLINT
                      )
RETURNS TABLE ( W_TAX    INTEGER
                , C_DISCOUNT INTEGER
                , C_LAST   VARCHAR(16)
                , C_CREDIT CHAR(2)
                )
SPECIFIC NEW_WH
MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL
ACTION LANGUAGE SQL
VAR: BEGIN ATOMIC
DECLARE C_DISCOUNT INTEGER;
DECLARE C_LAST   VARCHAR(16);
DECLARE C_CREDIT CHAR(2);
DECLARE W_TAX    INTEGER;
INSERT
INTO NEW_ORDER ( NO_O_ID, NO_D_ID, NO_W_ID )
VALUES ( O_ID
        , D_ID
        , W_ID
        )
;
INSERT
INTO ORDERS ( O_C_ID, O_ENTRY_D, O_CARRIER_ID,
O_OL_CNT, O_ALL_LOCAL, O_ID, O_W_ID, O_D_ID )
VALUES ( C_ID
        , O_ENTRY_D
        , 0
        , O_OL_CNT
        , O_ALL_LOCAL
        , O_ID
        , W_ID
        , D_ID
        )
;
SET ( C_DISCOUNT, C_LAST, C_CREDIT )
= ( SELECT C_DISCOUNT, C_LAST, C_CREDIT
    FROM CUSTOMER
    WHERE C_ID = NEW_WH.C_ID

```

```

AND C_W_ID = W_ID
AND C_D_ID = D_ID
)
;
SET W_TAX
= ( SELECT W_TAX
    FROM WAREHOUSE
    WHERE W_ID = NEW_WH.W_ID
    )
;
RETURN VALUES ( W_TAX , C_DISCOUNT , C_LAST ,
C_CREDIT );
END
$

```

Src.Srv/cat-proc.ddl

```

CREATE PROCEDURE news
(in new_in  varchar(270) FOR BIT DATA,
 out new_out varchar(662) FOR BIT DATA)
LANGUAGE C
PARAMETER STYLE GENERAL
EXTERNAL NAME
'/home/tpcc/sql/lib/function/news!news'
not fenced;
CREATE PROCEDURE ords
(in ord_in  varchar(42) FOR BIT DATA,
 out ord_out varchar(446) FOR BIT DATA)
LANGUAGE C
PARAMETER STYLE GENERAL
EXTERNAL NAME
'/home/tpcc/sql/lib/function/ords!ords'
not fenced;
CREATE PROCEDURE dels
(in del_in  varchar(22) FOR BIT DATA,
 out del_out varchar(50) FOR BIT DATA)
LANGUAGE C
PARAMETER STYLE GENERAL
EXTERNAL NAME
'/home/tpcc/sql/lib/function/dels!dels'
not fenced;

```

Src.Srv/dels.exp

```

#! Export file
dels

```

Src.Srv/news.exp

```

#! Export file
news

```

Src.Srv/ords.exp

```

#! Export file
ords

```

Src.Srv/tpcc_all_sql.sqc

```

/*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp.
1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
*****/

/*
 * tpcc_all_sql.sqc - Client/Server code for TPCC
 */

#include <stdlib.h>
#include <errno.h>
#include "db2tpcc.h"
#include "tpccapp.h"
#include "tpccdbg.h"

#include "sqlca.h"
#include "sql.h"

#include "lval.h"

// -----
// New Order SERVER
// -----

int static is_ORIGINAL( char *string, short length );

SQL_API_RC new_order_internal( char *pin, char *pout )
{
    struct out_neword_struct *neword;

    struct in_neword_struct *in_neword;

    struct sqlca sqlca ;

    int fbadItemDetected = 0 ;

    EXEC SQL BEGIN DECLARE SECTION;

    char c_last [ 16 ];
    char c_credit [ 2 ];
    sqlint32 c_discount ;

```



```

sqlint32 dist_tax ;
sqlint32 ware_tax ;

sqlint32 w_id ;
short d_id ;
sqlint32 c_id ;

sqlint32 next_o_id ;

short s_quantity ;

sqlint32 supply_w_id ;

short inputItemCount ;

char stockDistrictInformation [ 24 ] ;
char item_name[ 24 ] ;

sqlint64 o_entry_d ;

short allLocal ;

sqlint32 item_price ;

struct i_data_type { short len ; char data[ 50 ] ; } i_data ;
struct s_data_type { short len ; char data[ 50 ] ; } s_data ;

sqlint32 id0, id1, id2, id3, id4, id5, id6, id7 ;
sqlint32 id8, id9, id10, id11, id12, id13, id14 ;

sqlint32 supply_w_id0, supply_w_id1, supply_w_id2,
supply_w_id3 ;
sqlint32 supply_w_id4, supply_w_id5, supply_w_id6,
supply_w_id7 ;
sqlint32 supply_w_id8, supply_w_id9, supply_w_id10,
supply_w_id11 ;
sqlint32 supply_w_id12, supply_w_id13, supply_w_id14 ;

short ol_quantity0, ol_quantity1, ol_quantity2,
ol_quantity3 ;
short ol_quantity4, ol_quantity5, ol_quantity6,
ol_quantity7 ;
short ol_quantity8, ol_quantity9, ol_quantity10,
ol_quantity11 ;
short ol_quantity12, ol_quantity13, ol_quantity14 ;

EXEC SQL END DECLARE SECTION ;

int storedProcRc ;
int inputItemArrayIndex ;

char stockDistrictInformationArray [15][25] ;

#define stockDistrictInformation
stockDistrictInformationArray[ inputItemArrayIndex ]

// Redirected input fields

#define w_id in_neword->s_W_ID
#define d_id in_neword->s_D_ID
#define c_id in_neword->s_C_ID

```

```

#define o_entry_d in_neword->s_O_ENTRY_D_time

#define inputItemCount in_neword->s_O_OL_CNT

#define allLocal in_neword->s_all_local

// Redirected output fields

#define c_last neword->s_C_LAST
#define c_credit neword->s_C_CREDIT
#define c_discount neword->s_C_DISCOUNT
#define ware_tax neword->s_W_TAX
#define dist_tax neword->s_D_TAX
#define s_quantity neword-
>item[ inputItemArrayIndex ].s_S_QUANTITY

// This output field becomes an input field to order_line

#define next_o_id neword->s_O_ID

#define item_name neword-
>item[ inputItemArrayIndex ].s_I_NAME

// item_price holds the integer version of this value. If the
return structure was
// an integer this would not be necessary.

sqlint32 i_priceArray[ 15 ] ;

#define item_price i_priceArray[ inputItemArrayIndex ]

// Handle the generic/brand distinction

struct i_data_type i_dataArray[ 15 ] ;
struct s_data_type s_dataArray[ 15 ] ;

#define i_data i_dataArray[ inputItemArrayIndex ]
#define s_data s_dataArray[ inputItemArrayIndex ]

// Redirect hostvars to input structure

#define id0 in_neword->in_item[0].s_OL_I_ID
#define id1 in_neword->in_item[1].s_OL_I_ID
#define id2 in_neword->in_item[2].s_OL_I_ID
#define id3 in_neword->in_item[3].s_OL_I_ID
#define id4 in_neword->in_item[4].s_OL_I_ID
#define id5 in_neword->in_item[5].s_OL_I_ID
#define id6 in_neword->in_item[6].s_OL_I_ID
#define id7 in_neword->in_item[7].s_OL_I_ID
#define id8 in_neword->in_item[8].s_OL_I_ID
#define id9 in_neword->in_item[9].s_OL_I_ID
#define id10 in_neword->in_item[10].s_OL_I_ID
#define id11 in_neword->in_item[11].s_OL_I_ID
#define id12 in_neword->in_item[12].s_OL_I_ID
#define id13 in_neword->in_item[13].s_OL_I_ID
#define id14 in_neword->in_item[14].s_OL_I_ID

#define ol_quantity0 in_neword-
>in_item[ 0 ].s_OL_QUANTITY

```

```

#define ol_quantity1 in_neword-
>in_item[ 1 ].s_OL_QUANTITY
#define ol_quantity2 in_neword-
>in_item[ 2 ].s_OL_QUANTITY
#define ol_quantity3 in_neword-
>in_item[ 3 ].s_OL_QUANTITY
#define ol_quantity4 in_neword-
>in_item[ 4 ].s_OL_QUANTITY
#define ol_quantity5 in_neword-
>in_item[ 5 ].s_OL_QUANTITY
#define ol_quantity6 in_neword-
>in_item[ 6 ].s_OL_QUANTITY
#define ol_quantity7 in_neword-
>in_item[ 7 ].s_OL_QUANTITY
#define ol_quantity8 in_neword-
>in_item[ 8 ].s_OL_QUANTITY
#define ol_quantity9 in_neword-
>in_item[ 9 ].s_OL_QUANTITY
#define ol_quantity10 in_neword-
>in_item[ 10 ].s_OL_QUANTITY
#define ol_quantity11 in_neword-
>in_item[ 11 ].s_OL_QUANTITY
#define ol_quantity12 in_neword-
>in_item[ 12 ].s_OL_QUANTITY
#define ol_quantity13 in_neword-
>in_item[ 13 ].s_OL_QUANTITY
#define ol_quantity14 in_neword-
>in_item[ 14 ].s_OL_QUANTITY

#define supply_w_id0 in_neword-
>in_item[ 0 ].s_OL_SUPPLY_W_ID
#define supply_w_id1 in_neword-
>in_item[ 1 ].s_OL_SUPPLY_W_ID
#define supply_w_id2 in_neword-
>in_item[ 2 ].s_OL_SUPPLY_W_ID
#define supply_w_id3 in_neword-
>in_item[ 3 ].s_OL_SUPPLY_W_ID
#define supply_w_id4 in_neword-
>in_item[ 4 ].s_OL_SUPPLY_W_ID
#define supply_w_id5 in_neword-
>in_item[ 5 ].s_OL_SUPPLY_W_ID
#define supply_w_id6 in_neword-
>in_item[ 6 ].s_OL_SUPPLY_W_ID
#define supply_w_id7 in_neword-
>in_item[ 7 ].s_OL_SUPPLY_W_ID
#define supply_w_id8 in_neword-
>in_item[ 8 ].s_OL_SUPPLY_W_ID
#define supply_w_id9 in_neword-
>in_item[ 9 ].s_OL_SUPPLY_W_ID
#define supply_w_id10 in_neword-
>in_item[ 10 ].s_OL_SUPPLY_W_ID
#define supply_w_id11 in_neword-
>in_item[ 11 ].s_OL_SUPPLY_W_ID
#define supply_w_id12 in_neword-
>in_item[ 12 ].s_OL_SUPPLY_W_ID
#define supply_w_id13 in_neword-
>in_item[ 13 ].s_OL_SUPPLY_W_ID
#define supply_w_id14 in_neword-
>in_item[ 14 ].s_OL_SUPPLY_W_ID

EXEC SQL DECLARE ISOL_Remote_1 CURSOR FOR

```

```

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1) , :id0 , :ol_quantity0 , :supply_w_id0 )
) AS X ( OL_NUMBER ,
I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER

```

```

, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_2 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1) , :id0 , :ol_quantity0 , :supply_w_id0 )
,
( SMALLINT( 2) , :id1 , :ol_quantity1 , :supply_w_id1 )
) AS X ( OL_NUMBER ,
I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER

```

```

, D_ID
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_3 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1) , :id0 , :ol_quantity0 , :supply_w_id0 )
,
( SMALLINT( 2) , :id1 , :ol_quantity1 , :supply_w_id1 )
) AS X ( OL_NUMBER ,
I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER

```

```

, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )
) AS X ( OL_NUMBER ,
I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID

```

```

, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_4 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 , :supply_w_id3 )
) AS X ( OL_NUMBER ,
I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)

```

```

) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_5 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

```

```

FROM ( SELECT :next_o_id as O_ID
        , :w_id AS W_ID
        , :d_id as D_ID
        , OL_NUMBER
        , I_ID
        , I_SUPPLY_W_ID
        , I_QTY
      FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 , :supply_w_id3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 , :supply_w_id4 )
      ) AS X ( OL_NUMBER ,
I_ID , I_QTY , I_SUPPLY_W_ID )
      ) AS ITEMLIST
      , TABLE( NEW_OL_ALL( I_ID
        , I_QTY
        , W_ID
        , I_SUPPLY_W_ID
        , O_ID
        , D_ID
      )
      ) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

```

```

SELECT O_ID
      , D_ID
      , W_ID
      , OL_NUMBER
      , I_ID
      , I_SUPPLY_W_ID
      , OL_DELIVERY_D
      , I_QTY
      , TOTAL_PRICE
      , OL_DIST_INFO
      , I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_6 CURSOR FOR
WITH DATA AS ( SELECT O_ID
      , D_ID
      , W_ID
      , OL_NUMBER
      , I_ID
      , I_SUPPLY_W_ID
      , O AS OL_DELIVERY_D
      , I_QTY
      , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
      , OL_DIST_INFO
      , I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
        , :w_id AS W_ID
        , :d_id as D_ID
        , OL_NUMBER
        , I_ID
        , I_SUPPLY_W_ID
        , I_QTY
      FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 , :supply_w_id3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 , :supply_w_id4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 , :supply_w_id5 )
      ) AS X ( OL_NUMBER ,
I_ID , I_QTY , I_SUPPLY_W_ID )
      ) AS ITEMLIST
      , TABLE( NEW_OL_ALL( I_ID

```

```

      , I_QTY
      , W_ID
      , I_SUPPLY_W_ID
      , O_ID
      , D_ID
      )
      ) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
      , D_ID
      , W_ID
      , OL_NUMBER
      , I_ID
      , I_SUPPLY_W_ID
      , OL_DELIVERY_D
      , I_QTY
      , TOTAL_PRICE
      , OL_DIST_INFO
      , I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_7 CURSOR FOR
WITH DATA AS ( SELECT O_ID
      , D_ID
      , W_ID
      , OL_NUMBER
      , I_ID
      , I_SUPPLY_W_ID

```

```

, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 , :supply_w_id3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 , :supply_w_id4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 , :supply_w_id5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 , :supply_w_id6 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST

, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, O_ID
, D_ID
) AS NEW_OL_ALL

WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO

) AS NEW_OL_ALL

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 , :supply_w_id3 )

```

```

, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA

) AS INS

;

EXEC SQL DECLARE ISOL_Remote_8 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 , :supply_w_id3 )

```

```

( SMALLINT( 5 ) , :id4 , :ol_quantity4 , :supply_w_id4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 , :supply_w_id5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 , :supply_w_id6 )
( SMALLINT( 8 ) , :id7 , :ol_quantity7 , :supply_w_id7 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST

, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, O_ID
, D_ID
) AS NEW_OL_ALL

WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO

) AS NEW_OL_ALL

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

```

```

FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_9 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, O AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 , :supply_w_id3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 , :supply_w_id4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 , :supply_w_id5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 , :supply_w_id6 )
( SMALLINT( 8 ) , :id7 , :ol_quantity7 , :supply_w_id7 )
( SMALLINT( 9 ) , :id8 , :ol_quantity8 , :supply_w_id8 )
) AS X ( OL_NUMBER ,
I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
) AS NEW_OL_ALL
) AS NEW_OL_ALL
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_10 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, O AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_10 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, O AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

```

```

WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_10 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, O AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

```

```

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 , :supply_w_id3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 , :supply_w_id4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 , :supply_w_id5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 , :supply_w_id6 )
( SMALLINT( 8 ) , :id7 , :ol_quantity7 , :supply_w_id7 )
( SMALLINT( 9 ) , :id8 , :ol_quantity8 , :supply_w_id8 )
( SMALLINT( 10 ) , :id9 , :ol_quantity9 , :supply_w_id9 )
) AS X ( OL_NUMBER ,
I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
) AS NEW_OL_ALL
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY

```

```

        , OL_AMOUNT
        , OL_DIST_INFO
    )
    INCLUDE ( I_PRICE INTEGER
             , I_NAME CHAR(24)
             , I_DATA VARCHAR(50)
             , S_DATA VARCHAR(50)
             , S_QUANTITY SMALLINT )
    SELECT O_ID
           , D_ID
           , W_ID
           , OL_NUMBER
           , I_ID
           , I_SUPPLY_W_ID
           , OL_DELIVERY_D
           , I_QTY
           , TOTAL_PRICE
           , OL_DIST_INFO
           , I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY
    FROM DATA
    ) AS INS
;
EXEC SQL DECLARE ISOL_Remote_11 CURSOR FOR
    WITH DATA AS ( SELECT O_ID
                     , D_ID
                     , W_ID
                     , OL_NUMBER
                     , I_ID
                     , I_SUPPLY_W_ID
                     , 0 AS OL_DELIVERY_D
                     , I_QTY
                     , (I_PRICE * I_QTY) AS TOTAL_PRICE
                     , OL_DIST_INFO
                     , I_PRICE, I_NAME, I_DATA,
S_QUANTITY
    FROM ( SELECT :next_o_id as O_ID
           , :w_id AS W_ID
           , :d_id as D_ID
           , OL_NUMBER
           , I_ID
           , I_SUPPLY_W_ID
           , I_QTY
    FROM Table( VALUES
    ( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
    ( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
    ( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )
    ( SMALLINT( 4 ) , :id3 , :ol_quantity3 , :supply_w_id3 )

```

```

    ( SMALLINT( 5 ) , :id4 , :ol_quantity4 , :supply_w_id4 )
    ( SMALLINT( 6 ) , :id5 , :ol_quantity5 , :supply_w_id5 )
    ( SMALLINT( 7 ) , :id6 , :ol_quantity6 , :supply_w_id6 )
    ( SMALLINT( 8 ) , :id7 , :ol_quantity7 , :supply_w_id7 )
    ( SMALLINT( 9 ) , :id8 , :ol_quantity8 , :supply_w_id8 )
    ( SMALLINT( 10 ) , :id9 , :ol_quantity9 , :supply_w_id9 )
    ( SMALLINT( 11 ) , :id10 , :ol_quantity10 , :supply_w_id10 )
    ) AS X ( OL_NUMBER ,
I_ID , I_QTY , I_SUPPLY_W_ID )
    ) AS ITEMLIST
    , TABLE( NEW_OL_ALL( I_ID
                        , I_QTY
                        , W_ID
                        , I_SUPPLY_W_ID
                        , O_ID
                        , D_ID
                        )
    ) AS NEW_OL_ALL
    WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
    )
    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
    FROM NEW TABLE ( INSERT INTO ORDER_LINE
    ( OL_O_ID
    , OL_D_ID
    , OL_W_ID
    , OL_NUMBER
    , OL_I_ID
    , OL_SUPPLY_W_ID
    , OL_DELIVERY_D
    , OL_QUANTITY
    , OL_AMOUNT
    , OL_DIST_INFO
    )
    INCLUDE ( I_PRICE INTEGER
             , I_NAME CHAR(24)
             , I_DATA VARCHAR(50)
             , S_DATA VARCHAR(50)
             , S_QUANTITY SMALLINT )
    SELECT O_ID
           , D_ID
           , W_ID
           , OL_NUMBER
           , I_ID

```

```

           , I_SUPPLY_W_ID
           , OL_DELIVERY_D
           , I_QTY
           , TOTAL_PRICE
           , OL_DIST_INFO
           , I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY
    FROM DATA
    ) AS INS
;
EXEC SQL DECLARE ISOL_Remote_12 CURSOR FOR
    WITH DATA AS ( SELECT O_ID
                     , D_ID
                     , W_ID
                     , OL_NUMBER
                     , I_ID
                     , I_SUPPLY_W_ID
                     , 0 AS OL_DELIVERY_D
                     , I_QTY
                     , (I_PRICE * I_QTY) AS TOTAL_PRICE
                     , OL_DIST_INFO
                     , I_PRICE, I_NAME, I_DATA,
S_QUANTITY
    FROM ( SELECT :next_o_id as O_ID
           , :w_id AS W_ID
           , :d_id as D_ID
           , OL_NUMBER
           , I_ID
           , I_SUPPLY_W_ID
           , I_QTY
    FROM Table( VALUES
    ( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
    ( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
    ( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )
    ( SMALLINT( 4 ) , :id3 , :ol_quantity3 , :supply_w_id3 )
    ( SMALLINT( 5 ) , :id4 , :ol_quantity4 , :supply_w_id4 )
    ( SMALLINT( 6 ) , :id5 , :ol_quantity5 , :supply_w_id5 )
    ( SMALLINT( 7 ) , :id6 , :ol_quantity6 , :supply_w_id6 )
    ( SMALLINT( 8 ) , :id7 , :ol_quantity7 , :supply_w_id7 )
    ( SMALLINT( 9 ) , :id8 , :ol_quantity8 , :supply_w_id8 )
    ( SMALLINT( 10 ) , :id9 , :ol_quantity9 , :supply_w_id9 )
    ( SMALLINT( 11 ) , :id10 , :ol_quantity10 , :supply_w_id10 )
    )

```

```

( SMALLINT( 12) , :id11 , :ol_quantity11 , :supply_w_id11
)
) AS X ( OL_NUMBER ,
, I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY
FROM DATA
) AS INS
;

```

```

EXEC SQL DECLARE ISOL_Remote_13 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, O AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1) , :id0 , :ol_quantity0 , :supply_w_id0 )
( SMALLINT( 2) , :id1 , :ol_quantity1 , :supply_w_id1 )
( SMALLINT( 3) , :id2 , :ol_quantity2 , :supply_w_id2 )
( SMALLINT( 4) , :id3 , :ol_quantity3 , :supply_w_id3 )
( SMALLINT( 5) , :id4 , :ol_quantity4 , :supply_w_id4 )
( SMALLINT( 6) , :id5 , :ol_quantity5 , :supply_w_id5 )
( SMALLINT( 7) , :id6 , :ol_quantity6 , :supply_w_id6 )
( SMALLINT( 8) , :id7 , :ol_quantity7 , :supply_w_id7 )
( SMALLINT( 9) , :id8 , :ol_quantity8 , :supply_w_id8 )
( SMALLINT( 10) , :id9 , :ol_quantity9 , :supply_w_id9 )
( SMALLINT( 11) , :id10 , :ol_quantity10 , :supply_w_id10 )
)
) AS X ( OL_NUMBER ,
I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY

```

```

, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_14 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, O AS OL_DELIVERY_D

```



```

, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 , :supply_w_id3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 , :supply_w_id4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 , :supply_w_id5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 , :supply_w_id6 )
( SMALLINT( 8 ) , :id7 , :ol_quantity7 , :supply_w_id7 )
( SMALLINT( 9 ) , :id8 , :ol_quantity8 , :supply_w_id8 )
( SMALLINT( 10 ) , :id9 , :ol_quantity9 , :supply_w_id9 )
( SMALLINT( 11 ) , :id10 , :ol_quantity10 , :supply_w_id10 )
( SMALLINT( 12 ) , :id11 , :ol_quantity11 , :supply_w_id11 )
( SMALLINT( 13 ) , :id12 , :ol_quantity12 , :supply_w_id12 )
( SMALLINT( 14 ) , :id13 , :ol_quantity13 , :supply_w_id13 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY , I_SUPPLY_W_ID )
AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
) AS NEW_OL_ALL
) AS NEW_OL_ALL

```

```

NULL WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL )

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Remote_15 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, O AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID

```

```

, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 , :supply_w_id3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 , :supply_w_id4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 , :supply_w_id5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 , :supply_w_id6 )
( SMALLINT( 8 ) , :id7 , :ol_quantity7 , :supply_w_id7 )
( SMALLINT( 9 ) , :id8 , :ol_quantity8 , :supply_w_id8 )
( SMALLINT( 10 ) , :id9 , :ol_quantity9 , :supply_w_id9 )
( SMALLINT( 11 ) , :id10 , :ol_quantity10 , :supply_w_id10 )
( SMALLINT( 12 ) , :id11 , :ol_quantity11 , :supply_w_id11 )
( SMALLINT( 13 ) , :id12 , :ol_quantity12 , :supply_w_id12 )
( SMALLINT( 14 ) , :id13 , :ol_quantity13 , :supply_w_id13 )
( SMALLINT( 15 ) , :id14 , :ol_quantity14 , :supply_w_id14 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY , I_SUPPLY_W_ID )
AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
) AS NEW_OL_ALL

WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)

```

```

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_1 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID

```

```

, I_QTY
FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY )
) AS ITEMLIST

, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_2 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY

```

```

) AS INS
;

EXEC SQL DECLARE ISOL_Local_2 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY
) AS X ( OL_NUMBER ,
I_ID , I_QTY )
) AS ITEMLIST

, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY

```

```

, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_3 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
,
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
,
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
,
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY
) AS ITEMLIST

, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_4 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID

```

```

, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
,
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
,
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
,
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY
) AS ITEMLIST

, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_4 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID

```

```

, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
,
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
,
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
,
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY
) AS ITEMLIST

, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)

```

```

, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT)

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA
) AS INS
;

EXEC SQL DECLARE ISOL_Local_5 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
,
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
,
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
,
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
,
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
,
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT)

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA
) AS INS
;

EXEC SQL DECLARE ISOL_Local_6 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
,
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
,
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
,
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
,
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
,
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

```

```

, I_QTY
, W_ID
, O_ID
, D_ID
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT)

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA
) AS INS
;

EXEC SQL DECLARE ISOL_Local_6 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
,
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
,
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
,
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
,
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
,
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

```

```

, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
,
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
,
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
,
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
,
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
,
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

```

```

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_7 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
,
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
,
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
,
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
,
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
,
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
,
( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
,
( SMALLINT( 8 ) , :id7 , :ol_quantity7 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_8 CURSOR FOR

WITH DATA AS ( SELECT O_ID

```

```

) AS X ( OL_NUMBER ,
I_ID , I_QTY
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_8 CURSOR FOR

WITH DATA AS ( SELECT O_ID

```

```

, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
,
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
,
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
,
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
,
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
,
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
,
( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
,
( SMALLINT( 8 ) , :id7 , :ol_quantity7 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID

```

```

, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_9 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )

```

```

,
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
,
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
,
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
,
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
,
( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
,
( SMALLINT( 8 ) , :id7 , :ol_quantity7 )
,
( SMALLINT( 9 ) , :id8 , :ol_quantity8 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY )
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL )

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D

```

```

, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_10 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
( SMALLINT( 8 ) , :id7 , :ol_quantity7 )
( SMALLINT( 9 ) , :id8 , :ol_quantity8 )
( SMALLINT( 10 ) , :id9 , :ol_quantity9 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY )
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY

```

```

        , W_ID
        , O_ID
        , D_ID
    )
    ) AS NEW_OL_LOCAL

    WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_11 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
)

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_11 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY

```

```

        , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
        , OL_DIST_INFO
        , I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
,
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
,
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
,
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
,
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
,
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
,
( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
,
( SMALLINT( 8 ) , :id7 , :ol_quantity7 )
,
( SMALLINT( 9 ) , :id8 , :ol_quantity8 )
,
( SMALLINT( 10 ) , :id9 , :ol_quantity9 )
,
( SMALLINT( 11 ) , :id10 , :ol_quantity10 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY )
) AS ITEMLIST

, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID

```

```

        , OL_NUMBER
        , OL_I_ID
        , OL_SUPPLY_W_ID
        , OL_DELIVERY_D
        , OL_QUANTITY
        , OL_AMOUNT
        , OL_DIST_INFO
    )

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_12 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
,
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )

```

```

( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
( SMALLINT( 8 ) , :id7 , :ol_quantity7 )
( SMALLINT( 9 ) , :id8 , :ol_quantity8 )
( SMALLINT( 10 ) , :id9 , :ol_quantity9 )
( SMALLINT( 11 ) , :id10 , :ol_quantity10 )
( SMALLINT( 12 ) , :id11 , :ol_quantity11 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY )
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID

```

```

, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_13 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
( SMALLINT( 8 ) , :id7 , :ol_quantity7 )
( SMALLINT( 9 ) , :id8 , :ol_quantity8 )
( SMALLINT( 10 ) , :id9 , :ol_quantity9 )

```

```

( SMALLINT( 11 ) , :id10 , :ol_quantity10 )
( SMALLINT( 12 ) , :id11 , :ol_quantity11 )
( SMALLINT( 13 ) , :id12 , :ol_quantity12 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY )
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA

```



```

) AS INS
;
EXEC SQL DECLARE ISOL_Local_14 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
( SMALLINT( 8 ) , :id7 , :ol_quantity7 )
( SMALLINT( 9 ) , :id8 , :ol_quantity8 )
( SMALLINT( 10 ) , :id9 , :ol_quantity9 )
( SMALLINT( 11 ) , :id10 , :ol_quantity10 )
( SMALLINT( 12 ) , :id11 , :ol_quantity11 )
( SMALLINT( 13 ) , :id12 , :ol_quantity12 )
( SMALLINT( 14 ) , :id13 , :ol_quantity13 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)
) AS INS
;
EXEC SQL DECLARE ISOL_Local_15 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY

```

```

, W_ID
, O_ID
, D_ID
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA

) AS INS
;
EXEC SQL DECLARE ISOL_Local_15 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY

```

```

, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
( SMALLINT( 8 ) , :id7 , :ol_quantity7 )
( SMALLINT( 9 ) , :id8 , :ol_quantity8 )
( SMALLINT( 10 ) , :id9 , :ol_quantity9 )
( SMALLINT( 11 ) , :id10 , :ol_quantity10 )
( SMALLINT( 12 ) , :id11 , :ol_quantity11 )
( SMALLINT( 13 ) , :id12 , :ol_quantity12 )
( SMALLINT( 14 ) , :id13 , :ol_quantity13 )
( SMALLINT( 15 ) , :id14 , :ol_quantity14 )

) AS X ( OL_NUMBER ,
I_ID , I_QTY
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)
) AS INS
;
EXEC SQL DECLARE ISOL_Local_15 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY

```

```

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY

FROM DATA

) AS INS
;

// Start processing

in_neword = (struct in_neword_struct *) pin ;
neword = (struct out_neword_struct *) pout ;

#ifdef DEBUGIT
new_debug( neword, in_neword, "SP upon entry");
#endif

// Using I_PRICE == 0 as a flag to the client that the ITEM
was not fetched (hence bad).

for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
in_neword->s_O_OL_CNT ; inputItemArrayIndex++ )
{
i_priceArray[ inputItemArrayIndex ] = 0 ;
}

neword->deadlocks = -1 ;

```

```

retry_tran:

neword->deadlocks++ ;

EXEC SQL

SELECT D_TAX, D_NEXT_O_ID
INTO :dist_tax , :next_o_id

FROM OLD TABLE ( UPDATE DISTRICT

1 SET D_NEXT_O_ID = D_NEXT_O_ID +

WHERE D_W_ID = :w_id
AND D_ID = :d_id

) AS OT
;

if ( sqlca.sqlcode != 0 )
{
DLCHK( retry_tran );
sqlerror( NEWORD_SQL, "DISTRICT", __FILE__,
__LINE__, &sqlca ) ;
goto ferror;
}

// Invalid I_ID will give a +100, now that we've changed the
cursor definitions
// to include a 'WHERE I_PRICE NOT NULL' clause.

#define NEW_CURSOR_OPEN_ERROR
\
{
if( sqlca.sqlcode != 0 )
\
{
goto sql_error ;
\
}
\
}

#define NEW_CURSOR_ERROR
\
{
if( sqlca.sqlcode == 0 )
\
{
neword->s_O_OL_CNT ++ ;
\
}
\
}
else
\
if( sqlca.sqlcode == +100 )
\
{
break ;
\
}
else
\
goto sql_error ;
\
}

if ( allLocal )
{
switch( inputItemCount )
{
case 1:

```

```

EXEC SQL OPEN ISOL_Local_1 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Local_1
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 2:
EXEC SQL OPEN ISOL_Local_2 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Local_2
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 3:
EXEC SQL OPEN ISOL_Local_3 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Local_3
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 4:
EXEC SQL OPEN ISOL_Local_4 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Local_4
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 5:
EXEC SQL OPEN ISOL_Local_5 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Local_5
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 6:
EXEC SQL OPEN ISOL_Local_6 ;
NEW_CURSOR_OPEN_ERROR

```

```

        for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
        {
            EXEC SQL FETCH ISOL_Local_6
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
            NEW_CURSOR_ERROR
        }
        break;
    case 7:
        EXEC SQL OPEN ISOL_Local_7;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
        {
            EXEC SQL FETCH ISOL_Local_7
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
            NEW_CURSOR_ERROR
        }
        break;
    case 8:
        EXEC SQL OPEN ISOL_Local_8;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
        {
            EXEC SQL FETCH ISOL_Local_8
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
            NEW_CURSOR_ERROR
        }
        break;
    case 9:
        EXEC SQL OPEN ISOL_Local_9;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
        {
            EXEC SQL FETCH ISOL_Local_9
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
            NEW_CURSOR_ERROR
        }
        break;
    case 10:
        EXEC SQL OPEN ISOL_Local_10;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
        {
            EXEC SQL FETCH ISOL_Local_10
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
            NEW_CURSOR_ERROR
        }
        break;
    case 11:
        EXEC SQL OPEN ISOL_Local_11;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)

```

```

        {
            EXEC SQL FETCH ISOL_Local_11
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
            NEW_CURSOR_ERROR
        }
        break;
    case 12:
        EXEC SQL OPEN ISOL_Local_12;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
        {
            EXEC SQL FETCH ISOL_Local_12
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
            NEW_CURSOR_ERROR
        }
        break;
    case 13:
        EXEC SQL OPEN ISOL_Local_13;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
        {
            EXEC SQL FETCH ISOL_Local_13
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
            NEW_CURSOR_ERROR
        }
        break;
    case 14:
        EXEC SQL OPEN ISOL_Local_14;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
        {
            EXEC SQL FETCH ISOL_Local_14
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
            NEW_CURSOR_ERROR
        }
        break;
    case 15:
        EXEC SQL OPEN ISOL_Local_15;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
        {
            EXEC SQL FETCH ISOL_Local_15
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
            NEW_CURSOR_ERROR
        }
        break;
    default:
        sqlerror(NEWORD_SQL, "Default switch on local
orderline/stock/index", __FILE__, __LINE__, &sqlca);
        goto ferror;
    }
}

```

```

else
{
    switch( inputItemCount )
    {
        case 1:
            EXEC SQL OPEN ISOL_Remote_1;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
            {
                EXEC SQL FETCH ISOL_Remote_1
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
                NEW_CURSOR_ERROR
            }
            break;
        case 2:
            EXEC SQL OPEN ISOL_Remote_2;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
            {
                EXEC SQL FETCH ISOL_Remote_2
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
                NEW_CURSOR_ERROR
            }
            break;
        case 3:
            EXEC SQL OPEN ISOL_Remote_3;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
            {
                EXEC SQL FETCH ISOL_Remote_3
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
                NEW_CURSOR_ERROR
            }
            break;
        case 4:
            EXEC SQL OPEN ISOL_Remote_4;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
            {
                EXEC SQL FETCH ISOL_Remote_4
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
                NEW_CURSOR_ERROR
            }
            break;
        case 5:
            EXEC SQL OPEN ISOL_Remote_5;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
            {
                EXEC SQL FETCH ISOL_Remote_5
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
                NEW_CURSOR_ERROR
            }

```

```

    }
    break ;
case 6:
EXEC SQL OPEN ISOL_Remote_6 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Remote_6
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 7:
EXEC SQL OPEN ISOL_Remote_7 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Remote_7
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 8:
EXEC SQL OPEN ISOL_Remote_8 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Remote_8
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 9:
EXEC SQL OPEN ISOL_Remote_9 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Remote_9
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 10:
EXEC SQL OPEN ISOL_Remote_10 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Remote_10
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;

```

```

case 11:
EXEC SQL OPEN ISOL_Remote_11 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Remote_11
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 12:
EXEC SQL OPEN ISOL_Remote_12 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Remote_12
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 13:
EXEC SQL OPEN ISOL_Remote_13 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Remote_13
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 14:
EXEC SQL OPEN ISOL_Remote_14 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Remote_14
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
case 15:
EXEC SQL OPEN ISOL_Remote_15 ;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
{
EXEC SQL FETCH ISOL_Remote_15
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
NEW_CURSOR_ERROR
}
break ;
default:

```

```

sqlerror(NEWORD_SQL, "Default switch on remote
orderline/stock/index", __FILE__, __LINE__, &sqlca);
goto ferror;
}
}

for ( inputItemArrayIndex = 0 ;
inputItemArrayIndex < in_neword->s_O_OL_CNT //
from input
&& i_priceArray[ inputItemArrayIndex ] != 0 ;
inputItemArrayIndex++ )
{
// s_I_NAME, and s_S_QUANTITY already set as output
host variables

neword->item[ inputItemArrayIndex ].s_I_PRICE =
i_priceArray[ inputItemArrayIndex ] ;

if
( is_ORIGINAL( s_dataArray[ inputItemArrayIndex ].data,
s_dataArray[ inputItemArrayIndex ].len )
&&
is_ORIGINAL( i_dataArray[ inputItemArrayIndex ].data,
i_dataArray[ inputItemArrayIndex ].len ) )
{
neword->item[ inputItemArrayIndex ].s_brand_generic =
'B';
}
else
{
neword->item[ inputItemArrayIndex ].s_brand_generic =
'G';
}
}

EXEC SQL

SELECT W_TAX, C_DISCOUNT, C_LAST, C_CREDIT

INTO :ware_tax, :c_discount, :c_last, :c_credit

FROM TABLE ( NEW_WH ( :next_o_id
, :w_id
, :d_id
, :c_id
, :o_entry_d
, :inputItemCount
, :allLocal
)
) AS NEW_WH_TABLE
;

if ( sqlca.sqlcode == 0 )
{
if ( neword->s_O_OL_CNT == in_neword->s_O_OL_CNT )
{
neword->s_transtatus = TRAN_OK ;
EXEC SQL COMMIT;
}
if( sqlca.sqlcode != 0 )
{

```

```

        sqlerror(NEWWORD_SQL, "COMMIT", __FILE__,
__LINE__, &sqlca );
        goto ferror;
    }
}
else
{
    newword->s_transtatus = INVALID_ITEM ;

    EXEC SQL ROLLBACK WORK ;

    if ( sqlca.sqlcode != 0 )
    {
        newword->s_transtatus = FATAL_SQLERROR;

        sqlerror(NEWWORD_SQL, "ROLLBACK FAILED
(INVALID ITEM)", __FILE__, __LINE__, &sqlca);
        // no point in ferror
    }
}
else
{
    DLCHK( retry_tran );

    sqlerror( NEWWORD_SQL, "NEW_WH", __FILE__,
__LINE__, &sqlca);
    goto ferror;
}

/*-----*/
/* Return to client */
/*-----*/

mexit:

if ( sqlca.sqlcode >= 0 )
{
    storedProcRc = SQLZ_HOLD_PROC ;
}
else
{
    storedProcRc = SQLZ_DISCONNECT_PROC ;
}

#ifdef DEBUGIT
    new_debug( newword, in_newword, "SP prior to return");
#endif

return ( storedProcRc );

sql_error:

{
    char tempstr[ 4096 ] ;

    DLCHK( retry_tran ) ;

    sprintf( tempstr,
"inputItemCnt=%d, :next_o_id=%d, :d_id=%d, :w_id=%d",
inputItemCnt, next_o_id, d_id, w_id ) ;

```

```

        sqlerror( NEWWORD_SQL, tempstr , __FILE__, __LINE__,
&sqlca ) ;
    }

ferror:

    newword->s_transtatus = FATAL_SQLERROR;

    EXEC SQL ROLLBACK WORK;

    if ( sqlca.sqlcode != 0 )
    {
        sqlerror( NEWWORD_SQL, "ROLLBACK FAILED", __FILE__,
__LINE__, &sqlca ) ;
    }

    goto mexit ;
}

/*
** A little function to search for the string "ORIGINAL" given a
string and
** it's length
*/
static unsigned char skip[256] = {8,8,8,8,8,8,8,8, /*0-9*/
    8,8,8,8,8,8,8,8, /*10-19*/
    8,8,8,8,8,8,8,8, /*20-29*/
    8,8,8,8,8,8,8,8, /*30-39*/
    8,8,8,8,8,8,8,8, /*40-49*/
    8,8,8,8,8,8,8,8, /*50-59*/
    8,8,8,8,1,8,8,8,8, /*60-69*/
    8,4,8,3,8,8,0,8,2,7, /*70-79*/
    8,8,6,8,8,8,8,8,8,8, /*80-89*/
    8,8,8,8,8,8,8,8,8,8, /*90-99*/
    8,8,8,8,8,8,8,8,8,8, /*100-109*/
    8,8,8,8,8,8,8,8,8,8, /*110-119*/
    8,8,8,8,8,8,8,8,8,8, /*120-129*/
    8,8,8,8,8,8,8,8,8,8, /*130-139*/
    8,8,8,8,8,8,8,8,8,8, /*140-149*/
    8,8,8,8,8,8,8,8,8,8, /*150-159*/
    8,8,8,8,8,8,8,8,8,8, /*160-169*/
    8,8,8,8,8,8,8,8,8,8, /*170-179*/
    8,8,8,8,8,8,8,8,8,8, /*180-189*/
    8,8,8,8,8,8,8,8,8,8, /*190-199*/
    8,8,8,8,8,8,8,8,8,8, /*200-209*/
    8,8,8,8,8,8,8,8,8,8, /*210-219*/
    8,8,8,8,8,8,8,8,8,8, /*220-229*/
    8,8,8,8,8,8,8,8,8,8, /*230-239*/
    8,8,8,8,8,8,8,8,8,8, /*240-249*/
    8,8,8,8,8,8,8,8,8,8, /*250-254*/
};

static int is_ORIGINAL( char *string, short length )
{
    char *cur_string;
    char *end_string;
    unsigned char *skips;
    int skip_dist;
    int result = 0;

    cur_string = string+7;
    end_string = string + length;

```

```

skips = skip;

while (cur_string < end_string)
{
    skip_dist = skips[*cur_string];
    while ( (skip_dist > 0) && (cur_string < end_string) )
    {
        skip_dist = skips[*cur_string += skip_dist];
    }

    if (cur_string >= end_string)
        goto exit;

    if ( cur_string[-4] != 'G' )
        goto noMatch;

    if ( memcmp( cur_string-7, "ORIGINAL", 8 ) == 0 )
    {
        result = 1;
        goto exit;
    }
}

noMatch:
    cur_string += 8;
} /* end while */

exit:
    return ( result );
}

//-----
// Order Status SERVER
//-----

#undef w_id
#undef d_id
#undef c_id_input
#undef o_id
#undef o_entry_d
#undef o_carrier_d
#undef c_id
#undef c_first
#undef c_middle
#undef c_last
#undef c_balance

SQL_API_RC order_status_internal( char *pin, char *pout )
{
    struct in_ordstat_struct * in_ordstat = (struct in_ordstat_struct
*) pin ;
    struct out_ordstat_struct * ordstat = (struct out_ordstat_struct
*) pout ;

    struct sqlca sqlca ;

    EXEC SQL BEGIN DECLARE SECTION;

    // From input values

    ###sqlint32 w_id ;
    ###short d_id;
    sqlint32 c_id_input ;

```

```

    struct s_data_type { short len ; char data[ 16 ] ; }
c_last_input ;

// From queries

// From initial query

sqlint32 o_id ;
###sqlint32 c_id ;
short o_carrier_id ;
###sqlint64 o_entry_d ;

char c_first[ 16 ] ;
char c_middle[ 2 ] ;
###char c_last[ 16 ] ;
sqlint64 c_balance ;

// From cursor

sqlint32 ol_i_id ;
sqlint32 ol_supply_w_id ;
short ol_quantity ;
sqlint32 ol_amount ;
sqlint64 ol_delivery_d ;

EXEC SQL END DECLARE SECTION;

// NOTE: this varchar would normally live inside the declare
section
// but this package already declared the same field higher up.
Need the field
// within this scope though.

###struct s_data_type { short len ; char data[ 16 ] ; }
c_last_input ;

int storedProcRc ;
int itemArrayIndex = 0 ;

#define w_id in_ordstat->s_W_ID ;
#define d_id in_ordstat->s_D_ID ;
#define c_id_input in_ordstat->s_C_ID
#define o_id ordstat->s_O_ID
#define o_entry_d ordstat->s_O_ENTRY_D_time
#define o_carrier_id ordstat->s_O_CARRIER_ID
#define c_id ordstat->s_C_ID
#define c_first ordstat->s_C_FIRST
#define c_middle ordstat->s_C_MIDDLE
#define c_last ordstat->s_C_LAST
#define c_balance ordstat->s_C_BALANCE

EXEC SQL DECLARE read_orderline_cur CURSOR FOR

SELECT OL_I_ID, OL_SUPPLY_W_ID, OL_QUANTITY,
OL_AMOUNT, OL_DELIVERY_D

FROM ORDER_LINE

WHERE OL_W_ID = :w_id
AND OL_D_ID = :d_id
AND OL_O_ID = :o_id

```

```

FOR FETCH ONLY ;

ordstat->deadlocks = -1 ;

#ifdef DEBUGIT
ord_debug(ordstat, in_ordstat, "SP upon entry");
#endif

retry_tran:

ordstat->deadlocks ++ ;

if ( c_id_input == 0 )
{
c_last_input.len = strlen( in_ordstat->s_C_LAST ) ;
memcpy( c_last_input.data , in_ordstat->s_C_LAST ,
c_last_input.len ) ;

EXEC SQL

SELECT O_ID, O_CARRIER_ID, O_ENTRY_D ,
C_BALANCE, C_FIRST, C_MIDDLE, C_ID

INTO :o_id, :o_carrier_id , :o_entry_d , :c_balance, :c_first, :c_m
iddle, :c_last

FROM TABLE ( ORD_C_LAST( :w_id
, :d_id
, :c_last_input
) AS ORD_C_LAST
;
}
else
{
EXEC SQL

SELECT O_ID, O_CARRIER_ID, O_ENTRY_D ,
C_BALANCE, C_FIRST, C_MIDDLE ,C_LAST

INTO :o_id, :o_carrier_id , :o_entry_d , :c_balance, :c_first, :c_m
iddle, :c_last

FROM TABLE ( ORD_C_ID( :w_id
, :d_id
, :c_id_input
) AS ORD_C_ID
;
}

if ( sqlca.sqlcode != 0 )
{
DLCHK( retry_tran );
sqlerror( ORDSTAT_SQL, "READ CUST and ORDERS",
__FILE__, __LINE__, &sqlca ) ;
goto ferror;
}

/*-----*/

```

```

/* Read ORDER_LINES */
/*-----*/

EXEC SQL OPEN read_orderline_cur ;

if ( sqlca.sqlcode != 0 )
{
DLCHK( retry_tran );
sqlerror(ORDSTAT_SQL, "OPEN CURSOR
read_orderline_cur", __FILE__, __LINE__, &sqlca ) ;
goto ferror;
}

itemArrayIndex = 0 ;
{
do
{
EXEC SQL FETCH read_orderline_cur

INTO :ol_i_id , :ol_supply_w_id , :ol_quantity , :ol_amount , :ol_
delivery_d ;

if ( sqlca.sqlcode == 0 )
{
ordstat->item[ itemArrayIndex ].s_OL_I_ID =
ol_i_id ;
ordstat->item[ itemArrayIndex ].s_OL_SUPPLY_W_ID
= ol_supply_w_id ;
ordstat->item[ itemArrayIndex ].s_OL_QUANTITY =
ol_quantity ;
ordstat->item[ itemArrayIndex ].s_OL_AMOUNT =
ol_amount ;
ordstat-
>item[ itemArrayIndex ].s_OL_DELIVERY_D_time =
ol_delivery_d ;

itemArrayIndex++;
}
else
if (sqlca.sqlcode < 0 )
{
DLCHK( retry_tran );
sqlerror( ORDSTAT_SQL, "FETCH CURSOR
read_orderline_cur", __FILE__, __LINE__, &sqlca ) ;
goto ferror;
}
}
while ( sqlca.sqlcode == 0 ) ;

ordstat->s_ol_cnt = itemArrayIndex ;

EXEC SQL COMMIT ;

if ( sqlca.sqlcode == 0 )
{
ordstat->s_transtatus = TRAN_OK ;
}
else
{
DLCHK( retry_tran );
}

```

```

        sqlerror(ORDSTAT_SQL, "COMMIT", __FILE__,
__LINE__, &sqlca);
        goto ferror ;
    }
mexit:

    if ( sqlca.sqlcode >= 0 )
    {
        storedProcRc = SQLZ_HOLD_PROC ;
    }
    else
    {
        storedProcRc = SQLZ_DISCONNECT_PROC ;
    }

#ifdef DEBUGIT
    ord_debug(ordstat, in_ordstat, "SP prior to return");
#endif

    return ( storedProcRc ) ;

ferror:

    ordstat->s_transtatus = FATAL_SQLERROR ;

    EXEC SQL ROLLBACK WORK ;

    if ( sqlca.sqlcode != 0 )
    {
        sqlerror(ORDSTAT_SQL, "ROLLBACK FAILED", __FILE__,
__LINE__, &sqlca);
    }

    goto mexit;
}

// -----
// Delivery SERVER
// -----

#undef d_id
#undef c_id
#undef w_id
#undef o_carrier_id
#undef ol_delivery_d

SQL_API_RC delivery_internal ( char * pin, char * pout )
{
    struct in_delivery_struct * in_delivery = (struct
in_delivery_struct *) pin ;
    struct out_delivery_struct * delivery = (struct
out_delivery_struct *) pout ;

    struct sqlca sqlca ;

    int storedProcRc ;

    short district_id ;
    sqlint32 customer_id ;

    EXEC SQL BEGIN DECLARE SECTION;

```

```

// input

    ##sqlint32 w_id ;
    ##short d_id ;
    ##sqlint32 c_id ;
    ##short o_carrier_id ;
    ##sqlint64 ol_delivery_d ;

// output

    short no_o_id_indicator = 0 ;
    sqlint32 no_o_id ;

EXEC SQL END DECLARE SECTION;

#define d_id district_id
#define c_id customer_id

#define w_id in_delivery->s_W_ID
#define o_carrier_id in_delivery->s_O_CARRIER_ID
#define ol_delivery_d in_delivery->s_O_DELIVERY_D_time

delivery->deadlocks = -1 ;

#ifdef DEBUGIT
    del_debug( delivery, in_delivery, "SP upon entry");
#endif

// Deadlock Handling
// -----
// Since we COMMIT inside the for() loop, we must take
special
// care while handling deadlocks. This is best explained by
// an example.
//
// Assume we deadlock on d_id=6. This means that an order
from the
// first 5 districts have already been delivered. We will then
// restart the loop (retry_tran). However, the loop will restart
// at d_id = 1! This means that the second (and all
subsequent)
// time through the loop, we will deliver orders for districts that
// have already been delivered, with the net result being more
than
// 10 orders being delivered.
//
// The solution to this problem is to initialize the starting point
// of the loop "before" the retry_tran label. This will ensure
that
// if we deadlock, we will restart the loop with the same district
// that we deadlocked on, and we won't deliver any extra
orders.
//
// NOTE: If we ever change this back to one COMMIT per
transaction
// (instead of one COMMIT per iteration), then the initialization
// of d_id must be moved back into the for loop. (A rollback
due
// to deadlock in this case would rollback all delivered orders
so
// far, so we'd need to re-deliver them all on the next iteration.)

```

```

d_id = 1;

retry_tran:

delivery->deadlocks++;

for ( ; d_id <= DISTRICTS_PER_WAREHOUSE ; d_id++ )
{
    no_o_id = 0 ;
    no_o_id_indicator = 0 ;

    EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC

        SELECT O_ID

            INTO :no_o_id :no_o_id_indicator

        FROM TABLE
( DEL( :w_id , :d_id , :o_carrier_id , :ol_delivery_d ) ) AS T ;

        COMMIT ;

    END COMPOUND ;

    if ( sqlca.sqlcode == 0 )
    {
        /* Refer to clause 2.7.4.2, bullet 3 in spec.*/
        /* Need to report if more than 1 or 1% of */
        /* no_o_id will remain 0 if null returned, so just treat the
same way */

        delivery->s_O_ID[ d_id - 1 ] = no_o_id ;
    }
    else
    {
        DLCHK( retry_tran );

        sqlerror( DELIVERY_SQL , "DELIVERY", __FILE__,
__LINE__, &sqlca);
        goto ferror ;
    }
}

delivery->s_transtatus = TRAN_OK ;

mexit:

    if ( sqlca.sqlcode >= 0 )
    {
        storedProcRc = SQLZ_HOLD_PROC ;
    }
    else
    {
        storedProcRc = SQLZ_DISCONNECT_PROC ;
    }

#ifdef DEBUGIT
    del_debug( delivery, in_delivery, "SP prior to return");
#endif

    return ( storedProcRc ) ;

```

```

ferror:

delivery->s_transtatus = FATAL_SQLERROR ;

EXEC SQL ROLLBACK WORK ;

if ( sqlca.sqlcode != 0 )
{
  sqlerror( DELIVERY_SQL, "ROLLBACK FAILED", __FILE__,
  __LINE__, &sqlca ) ;
}

goto mexit ;
}

// -----
// Stored Procedure Stubs
// -----

SQL_API_RC SQL_API_FN news( char *pin, char *pout )
{
  return new_order_internal( pin, pout ) ;
}

SQL_API_RC SQL_API_FN ords( char *pin, char *pout )
{
  return order_status_internal( pin, pout ) ;
}

SQL_API_RC SQL_API_FN dels ( char * pin, char * pout )
{
  return delivery_internal( pin, pout ) ;
}

```

Src.Srv/uncat-func.ddl

```

-----
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996
-- 2006
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with
-- IBM Corp.
-----
-- uncat-func.ddl - Drop table function DDL
--

-- STOCK LEVEL
DROP SPECIFIC FUNCTION STOCK_LEVEL $
-- DELIVERY
DROP SPECIFIC FUNCTION DELIVERY $
-- ORDER STATUS

```

```

DROP SPECIFIC FUNCTION ORD_C_LAST $
DROP SPECIFIC FUNCTION ORD_C_ID $
-- PAYMENT
DROP SPECIFIC FUNCTION PAY_C_LAST $
DROP SPECIFIC FUNCTION PAY_C_ID $
-- NEW ORDER
DROP SPECIFIC FUNCTION NEW_OL_ALL $
DROP SPECIFIC FUNCTION NEW_OL_LOCAL $
DROP SPECIFIC FUNCTION NEW_WH $

```

Src.Srv/uncat-proc.ddl

```

DROP PROCEDURE news
      (varchar(270),varchar(662));

DROP PROCEDURE news;

DROP PROCEDURE pays;

DROP PROCEDURE ords
      (varchar(42),varchar(446));

DROP PROCEDURE ords;

DROP PROCEDURE dels
      (varchar(22),varchar(50));

DROP PROCEDURE dels;

DROP PROCEDURE stks
      (varchar(18),varchar(14));

DROP PROCEDURE stks;

```

include/db2tpcc.h

```

/*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp.
** 1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
** IBM Corp.
*****/

/*
 * db2tpcc.h - Macros and Miscellany
 */

```

```

#ifndef __DB2TPCC_H
#define __DB2TPCC_H

#include <sys/types.h>

#include "lval.h"

/*
*****
*/
/* Transaction Return Codes (s_transtatus)
*/
/*
*****
*/

#define INVALID_ITEM      100
#define TRAN_OK          0
#define FATAL_SQLERROR   -1

/*
*****
*/
/* Definition of Unused and Bad Items
*/
/*
*****
*/
/* Define unused item ID to be 0. This allows the SUT to
determine the
*/
/* number of items in the order as required by 2.4.1.3 and
2.4.2.2 since
*/
/* the assumption that any item with OL_I_ID = 0 is unused will
be true.
*/
/* This in turn requires that the value used for an invalid item is
*/
/* equal to ITEMS + 1.
*/
/*
*****
*/

#define INVALID_ITEM_ID (2 * ITEMS) + 1
#define UNUSED_ITEM_ID 0

#define MIN_WAREHOUSE 1
#define MAX_WAREHOUSE WAREHOUSES

/*****
***
*/
/* NURand Constants
*/
/* C_C_LAST_RUN and C_C_LAST_LOAD must adhere to
clause 2.1.6.
*/
/* Analysis indicates that a C_LAST delta of 85 is optimal.
*/
/*****
***
*/
#define C_C_LAST_RUN      88
#define C_C_LAST_LOAD    173
#define C_C_ID            319
#define C_OL_I_ID        3849
#define A_C_LAST         255
#define A_C_ID           1023
#define A_OL_I_ID        8191

```



```

/*****
***
/* Transaction Type Identifiers */
/*****
***

#define CLIENT_SQL 0
#define NEWORD_SQL 1
#define PAYMENT_SQL 2
#define ORDSTAT_SQL 3
#define DELIVERY_SQL 4
#define STOCKLEV_SQL 5

#define SPGENERAL_PAD 3
#define SPGENERAL_ADJUST sizeof(int16_t)

struct in_neword_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    struct in_items_struct {
        int32_t s_OL_I_ID;
        int32_t s_OL_SUPPLY_W_ID;
        int16_t s_OL_QUANTITY;
        int16_t pad1[3];
    } in_item[15];
    int64_t s_O_ENTRY_D_time; /* init by SUT */
    int32_t s_C_ID;
    int32_t s_W_ID;
    int16_t s_D_ID;
    int16_t s_O_OL_CNT; /* init by SUT */
    int16_t s_all_local;
    int16_t duplicate_items;
};

struct out_neword_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    struct items_struct {
        int32_t s_I_PRICE;
        int32_t s_OL_AMOUNT;
        int16_t s_S_QUANTITY;
        int16_t pad2;
        char s_I_NAME[25];
        char s_brand_generic;
    } item[15];
    int64_t s_O_ENTRY_D_time;
    int32_t s_W_TAX;
    int32_t s_D_TAX;
    int32_t s_C_DISCOUNT;
    int32_t s_total_amount;
    int32_t s_O_ID;
    int16_t s_O_OL_CNT;
    int16_t s_transtatus;
    int16_t deadlocks;
    char s_C_LAST[17];
    char s_C_CREDIT[3];
};

struct in_payment_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];

```

```

    int64_t s_H_DATE_time; /* init by SUT */
    int64_t s_H_AMOUNT;
    int32_t s_W_ID;
    int32_t s_C_W_ID;
    int32_t s_C_ID;
    int16_t s_C_D_ID;
    int16_t s_D_ID;
    char s_C_LAST[17];
};

struct out_payment_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int64_t s_H_DATE_time;
    int64_t s_C_SINCE_time;
    int64_t s_C_CREDIT_LIM;
    int64_t s_C_BALANCE;
    int32_t s_C_DISCOUNT;
    int32_t s_C_ID;
    int16_t s_transtatus;
    int16_t deadlocks;
    char s_W_STREET_1[21];
    char s_W_STREET_2[21];
    char s_W_CITY[21];
    char s_W_STATE[3];
    char s_W_ZIP[10];
    char s_D_STREET_1[21];
    char s_D_STREET_2[21];
    char s_D_CITY[21];
    char s_D_STATE[3];
    char s_D_ZIP[10];
    char s_C_FIRST[17];
    char s_C_MIDDLE[3];
    char s_C_LAST[17];
    char s_C_STREET_1[21];
    char s_C_STREET_2[21];
    char s_C_CITY[21];
    char s_C_STATE[3];
    char s_C_ZIP[10];
    char s_C_PHONE[17];
    char s_C_CREDIT[3];
    char s_C_DATA[201];
};

struct in_ordstat_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int32_t s_C_ID;
    int32_t s_W_ID;
    int16_t s_D_ID;
    int16_t pad1[3];
    char s_C_LAST[17];
};

struct out_ordstat_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int64_t s_C_BALANCE;
    int64_t s_O_ENTRY_D_time;
    int32_t s_C_ID;
    int32_t s_O_ID;
    int16_t s_O_CARRIER_ID;

```

```

    int16_t s_ol_cnt;
    int16_t pad1[2];
    struct oitems_struct {
        int64_t s_OL_DELIVERY_D_time;
        int32_t s_OL_AMOUNT;
        int32_t s_OL_I_ID;
        int32_t s_OL_SUPPLY_W_ID;
        int16_t s_OL_QUANTITY;
        int16_t pad2;
    } item[15];
    int16_t s_transtatus;
    int16_t deadlocks;
    char s_C_FIRST[17];
    char s_C_MIDDLE[3];
    char s_C_LAST[17];
};

struct in_delivery_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int64_t s_O_DELIVERY_D_time; /* init by SUT */
    int32_t s_W_ID;
    int16_t s_O_CARRIER_ID;
};

struct out_delivery_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int32_t s_O_ID[10];
    int16_t s_transtatus;
    int16_t deadlocks;
};

struct in_stocklev_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int32_t s_threshold;
    int32_t s_W_ID;
    int16_t s_D_ID;
};

struct out_stocklev_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int32_t s_low_stock;
    int16_t s_transtatus;
    int16_t deadlocks;
};

/*
*****
/* Transaction Prototypes */
/*
*****
*/

#ifdef __cplusplus
extern "C" {
#endif

```

```
extern int neword_sql(struct in_neword_struct*, struct
out_neword_struct*);
extern int payment_sql(struct in_payment_struct*, struct
out_payment_struct*);
extern int ordstat_sql(struct in_ordstat_struct*, struct
out_ordstat_struct*);
extern int delivery_sql(struct in_delivery_struct*, struct
out_delivery_struct*);
extern int stocklev_sql(struct in_stocklev_struct*, struct
out_stocklev_struct*);
```

```
#ifdef __cplusplus
}
#endif
```

```
/*
*****
*/
/* DB2 Connect/Disconnect & Thread Context Wrappers
*/
/*
*****
*/
```

```
#ifdef __cplusplus
extern "C" {
#endif
```

```
extern int connect_to_TM(char*);
extern int connect_to_TM_auth(char*, char*, char*);
extern int disconnect_from_TM(void);
```

```
#ifdef __cplusplus
}
#endif
```

```
#endif // __DB2TPCC_H
```

include/lval.h

```
#ifndef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 81600
#define DISTRICTS_PER_WAREHOUSE 10
#define CUSTOMERS_PER_DISTRICT 3000
#define ITEMS 100000
#define STOCK_PER_WAREHOUSE 100000
#define MIN_OL_PER_ORDER 5
#define MAX_OL_PER_ORDER 15
#define NU_ORDERS_PER_DISTRICT 900
#endif // __LVAL_H
```

include/tpccapp.h

```
/*
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
```

```
**
** (C) COPYRIGHT International Business Machines Corp.
1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
*****
****/
```

```
/*
* tpccapp.h - Application Macros
*/
```

```
#ifndef __TPCCAPP_H
#define __TPCCAPP_H
```

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <time.h>
```

```
#define daricall
```

```
#include "sqlca.h"
#include "sqlcodes.h"
```

```
#ifdef SWAP_ENDIAN
#define SWAP_BYTE(Var) SwapEndian((void*)&Var,
sizeof(Var))
```

```
/*
*****
*****
FUNCTION: SwapEndian
PURPOSE: Swap the byte order of a structure
EXAMPLE: int l=0x12345678; SWAP_BYTE(l); l =>
0x78563412;
IMPLEMENTATION: Fold Addr in half, swap header & tail by
XOR op
e.g.: *a = 0x12 [ Addr + 0];
      *b = 0x78 [ Addr + 4 - 0 - 1 = Addr+3];
      *a ^= *b;      // sets *a to 0x6A
      *b ^= *a;      // sets *b to 0x12
      *a ^= *b;      // sets *a to 0x78
```

```
      Now *a => 0x78 && *b => 0x12
*****/
```

```
void SwapEndian(void *Addr, int nb)
{
int i;
for (i=0; i<nb/2; i++)
{
char *a = (char*)Addr+i;
char *b = (char*)Addr+(nb-i-1);

*a ^= *b;
*b ^= *a;
*a ^= *b;
```

```
}
}
#endif //SWAP_ENDIAN
```

```
/*
*****
** SQLCODE Macros
*****/
```

```
#define DLCHK(a) \
if (sqlca.sqlcode == SQL_RC_E911) { goto a; }

#define NACOMPCHK(last) \
if (sqlca.sqlcode != SQL_RC_E1339) { last = -1; } \
else { int a = ((sqlca.sqlerrmc[4] == 0x20) ? 0 : \
sqlca.sqlerrmc[4]-0x30); \
int b = ((sqlca.sqlerrmc[5] == 0x20) ? 0 : sqlca.sqlerrmc[5]- \
0x30); \
if (b == 0) { last = a; } else { last = a * 10 + b; } \
}
```

```
#endif // __TPCCAPP_H
```

include/tpccdbg.h

```
/*
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp.
1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
*****/
```

```
/*
* tpccdbg.h - Debugging Macros
*/
```

```
#ifndef __TPCCDBG_H
#define __TPCCDBG_H
```

```
#ifdef __cplusplus
extern "C" {
#endif
```

```
extern void new_debug (struct out_neword_struct *neword_ptr,
struct in_neword_struct *in_neword_ptr,
char *msg);
extern void pay_debug (struct out_payment_struct
*payment_ptr,
struct in_payment_struct *in_payment_ptr,
```

```

char *msg);
extern void ord_debug (struct out_ordstat_struct *ordstat_ptr,
struct in_ordstat_struct *in_ordstat_ptr,
char *msg);
extern void del_debug (struct out_delivery_struct *delivery_ptr,
struct in_delivery_struct *in_delivery_ptr,
char *msg);
extern void stk_debug (struct out_stocklev_struct *stocklev_ptr,
struct in_stocklev_struct *in_stocklev_ptr,
char *msg);

extern void new_print (struct out_neword_struct *neword_ptr,
struct in_neword_struct *in_neword_ptr,
char *filename,
char *msg);
extern void pay_print (struct out_payment_struct *payment_ptr,
struct in_payment_struct *in_payment_ptr,
char *filename,
char *msg);
extern void ord_print (struct out_ordstat_struct *ordstat_ptr,
struct in_ordstat_struct *in_ordstat_ptr,
char *filename,
char *msg);
extern void del_print (struct out_delivery_struct *delivery_ptr,
struct in_delivery_struct *in_delivery_ptr,
char *filename,
char *msg);
extern void stk_print (struct out_stocklev_struct *stocklev_ptr,
struct in_stocklev_struct *in_stocklev_ptr,
char *filename,
char *msg);

#ifdef __cplusplus
}
#endif

#endif // __TPCCDBG_H

tpccenv.sh
#####
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp.
1996 - 2006
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication
or
## disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
#####
#####
#
# tpccenv.sh - UNIX Environment Setup
#

```

```

# The Kit Version
export TPCC_VERSION=CK040318

# The DB2 Instance Name (for DB2)
export DB2INSTANCE=${USER}

# The OS being used (i.e. "UNIX", "WINDOWS")
export PLATFORM=UNIX

# The type of make command and slash used by the OS.
# (i.e. UNIX - "/", WINDOWS - "\").
# These are referenced all over the kit.
export SLASH="/";
export MAKE=make

# Specifies whether or not to use dari stored proc's for the TPC-
C driver. Set to either DARIVERISION or NONDARI;
#export TPCC_SPTYPE=NOSP
#export TPCC_SPTYPE=SPGENERAL2
export TPCC_SPTYPE=SPGENERAL
#export TPCC_SPTYPE=DARI2SQLDA

export DB2VERSION=v8

# The schema name is typically the SQL authorization ID (or
username).
# This is required for runstats and EEE.
export TPCC_SCHEMA=${USER}

# DB2 EE/EEE Configuration
export DB2EDITION=EE
#export DB2EDITION=EEE
export DB2NODE=0
export DB2NODES=1; # set to the number of nodes you
have. Set to 1 for EE.

# TPCC General Configuration
export TPCC_DBNAME=TPCC
export TPCC_ROOT=${HOME}/tpc-c.ibm
export TPCC_SQLLIB=${HOME}/sqllib
export TPCC_RUNDATA=${HOME}/tpccdata

# TPCC Debug Configuration
# This is the path where all error and debug logs are placed.
# To get debugging from within the stored procedures, you must
# set DB2ENVLIST="TPCC_DEBUGDIR" in tpcc.config.
export TPCC_DEBUGDIR=/tmp

# Specifies where stored procedures should be placed and if
they should
# be fenced.
export TPCC_SPDIR=${TPCC_SQLLIB}/function
export TPCC_FENCED=NO

utils/EXPLAIN.DDL
-- *- sql *-
--
-- Sample DDL to create Explain tables for Version 5.0
--

```

```

-- -> assumes db2start issued
-- -> assumes connection to a database exists
-- -> assumes called by "db2 -tf EXPLAIN.DDL"
--
--
-- To remind users how to use this file!
--
ECHO ;
ECHO ***** IMPORTANT ***** ;
ECHO ;
ECHO USAGE: db2 -tf EXPLAIN.DDL ;
ECHO ;
ECHO ***** IMPORTANT ***** ;
ECHO ;
ECHO ;
--
-- Set autocommit off
--
UPDATE COMMAND OPTIONS USING C OFF;
--
-- EXPLAIN INSTANCE
--
-- (must be defined first due to referential integrity definitions)
--
CREATE TABLE EXPLAIN_INSTANCE
( EXPLAIN_REQUESTER VARCHAR(128) NOT NULL,
EXPLAIN_TIME TIMESTAMP NOT
NULL,
SOURCE_NAME VARCHAR(128) NOT
NULL,
SOURCE_SCHEMA VARCHAR(128)
NOT NULL,
SOURCE_VERSION VARCHAR(64)
NOT NULL,
EXPLAIN_OPTION CHAR(1) NOT
NULL,
SNAPSHOT_TAKEN CHAR(1) NOT
NULL,
DB2_VERSION CHAR(7) NOT NULL,
SQL_TYPE CHAR(1) NOT NULL,
QUERYOPT INTEGER NOT NULL,
BLOCK CHAR(1) NOT NULL,
ISOLATION CHAR(2) NOT NULL,
BUFFPAGE INTEGER NOT NULL,
AVG_APPLS INTEGER NOT NULL,
SORTHEAP INTEGER NOT NULL,
LOCKLIST INTEGER NOT NULL,
MAXLOCKS SMALLINT NOT NULL,
LOCKS_AVAIL INTEGER NOT NULL,
CPU_SPEED DOUBLE NOT NULL,
REMARKS VARCHAR(254),
DBHEAP INTEGER NOT NULL,
COMM_SPEED DOUBLE NOT
NULL,
PARALLELISM CHAR(2) NOT NULL,
DATAJOINER CHAR(1) NOT NULL,
PRIMARY KEY
(EXPLAIN_REQUESTER,
EXPLAIN_TIME,
SOURCE_NAME,

```

```

SOURCE_SCHEMA,
SOURCE_VERSION))
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_STATEMENT
--
CREATE TABLE EXPLAIN_STATEMENT
( EXPLAIN_REQUESTER VARCHAR(128) NOT NULL,
  EXPLAIN_TIME     TIMESTAMP NOT
NULL,
  SOURCE_NAME      VARCHAR(128) NOT
NULL,
  SOURCE_SCHEMA    VARCHAR(128)
NOT NULL,
  SOURCE_VERSION   VARCHAR(64)
NOT NULL,
  EXPLAIN_LEVEL   CHAR(1) NOT
NULL,
  STMTNO          INTEGER NOT NULL,
  SECTNO          INTEGER NOT NULL,
  QUERYNO         INTEGER NOT
NULL,
  QUERYTAG        CHAR(20) NOT
NULL,
  STATEMENT_TYPE  CHAR(2) NOT
NULL,
  UPDATABLE       CHAR(1) NOT
NULL,
  DELETABLE       CHAR(1) NOT NULL,
  TOTAL_COST      DOUBLE NOT
NULL,
  STATEMENT_TEXT  CLOB(2M) NOT
NULL NOT LOGGED,
  SNAPSHOT        BLOB(10M) NOT
LOGGED,
  QUERY_DEGREE    INTEGER NOT
NULL,
  PRIMARY KEY
(EXPLAIN_REQUESTER,
  EXPLAIN_TIME,
  SOURCE_NAME,
  SOURCE_SCHEMA,
  SOURCE_VERSION,
  EXPLAIN_LEVEL,
  STMTNO,
  SECTNO),
  FOREIGN KEY
(EXPLAIN_REQUESTER,
  EXPLAIN_TIME,
  SOURCE_NAME,
  SOURCE_SCHEMA,
  SOURCE_VERSION)
REFERENCES EXPLAIN_INSTANCE
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_ARGUMENTS
--
CREATE TABLE EXPLAIN_ARGUMENT
( EXPLAIN_REQUESTER VARCHAR(128) NOT NULL,

```

```

EXPLAIN_TIME     TIMESTAMP NOT
NULL,
SOURCE_NAME      VARCHAR(128)
NOT NULL,
SOURCE_SCHEMA    VARCHAR(128)
NOT NULL,
SOURCE_VERSION   VARCHAR(64)
NOT NULL,
EXPLAIN_LEVEL   CHAR(1) NOT
NULL,
STMTNO          INTEGER NOT
NULL,
SECTNO          INTEGER NOT
NULL,
OPERATOR_ID     INTEGER NOT
NULL,
ARGUMENT_TYPE   CHAR(8) NOT
NULL,
ARGUMENT_VALUE  VARCHAR(1024),
LONG_ARGUMENT_VALUE CLOB(2M)
NOT LOGGED,
  FOREIGN KEY (EXPLAIN_REQUESTER,
EXPLAIN_TIME,
SOURCE_NAME,
SOURCE_SCHEMA,
SOURCE_VERSION,
EXPLAIN_LEVEL,
STMTNO,
SECTNO)
REFERENCES EXPLAIN_STATEMENT
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_OBJECT
--
CREATE TABLE EXPLAIN_OBJECT ( EXPLAIN_REQUESTER
VARCHAR(128) NOT NULL,
  EXPLAIN_TIME     TIMESTAMP NOT
NULL,
  SOURCE_NAME      VARCHAR(128) NOT
NULL,
  SOURCE_SCHEMA    VARCHAR(128)
NOT NULL,
  SOURCE_VERSION   VARCHAR(64)
NOT NULL,
  EXPLAIN_LEVEL   CHAR(1) NOT
NULL,
  STMTNO          INTEGER NOT NULL,
  SECTNO          INTEGER NOT NULL,
  OBJECT_SCHEMA   VARCHAR(128)
NOT NULL,
  OBJECT_NAME     VARCHAR(128) NOT
NULL,
  OBJECT_TYPE     CHAR(2) NOT
NULL,
  CREATE_TIME     TIMESTAMP,
  STATISTICS_TIME  TIMESTAMP,
  COLUMN_COUNT    SMALLINT NOT
NULL,
  ROW_COUNT       BIGINT NOT
NULL,

```

```

WIDTH           INTEGER NOT NULL,
PAGES           INTEGER NOT NULL,
DISTINCT        CHAR(1) NOT NULL,
TABLESPACE_NAME VARCHAR(128),
OVERHEAD        DOUBLE NOT
NULL,
TRANSFER_RATE   DOUBLE NOT
NULL,
PREFETCHSIZE    INTEGER NOT
NULL,
EXTENTS_SIZE    INTEGER NOT
NULL,
CLUSTER         DOUBLE NOT NULL,
NLEAF           INTEGER NOT NULL,
NLEVELS         INTEGER NOT NULL,
FULLKEYCARD     BIGINT NOT
NULL,
OVERFLOW        INTEGER NOT
NULL,
FIRSTKEYCARD    BIGINT NOT
NULL,
FIRST2KEYCARD   BIGINT NOT
NULL,
FIRST3KEYCARD   BIGINT NOT
NULL,
FIRST4KEYCARD   BIGINT NOT
NULL,
SEQUENTIAL_PAGES INTEGER NOT
NULL,
DENSITY         INTEGER NOT NULL,
STATS_SRC       CHAR(1) NOT
NULL,
AVERAGE_SEQUENCE_GAP
DOUBLE NOT NULL,
AVERAGE_SEQUENCE_FETCH_GAP
DOUBLE NOT NULL,
AVERAGE_SEQUENCE_PAGES
DOUBLE NOT NULL,
AVERAGE_SEQUENCE_FETCH_PAGES
DOUBLE NOT NULL,
AVERAGE_RANDOM_PAGES
DOUBLE NOT NULL,
AVERAGE_RANDOM_FETCH_PAGES
DOUBLE NOT NULL,
NUMRIDS         BIGINT NOT
NULL,
NUMRIDS_DELETED BIGINT NOT
NULL,
NUM_EMPTY_LEAFS BIGINT
NOT NULL,
ACTIVE_BLOCKS   BIGINT NOT
NULL,
  FOREIGN KEY (EXPLAIN_REQUESTER,
EXPLAIN_TIME,
SOURCE_NAME,
SOURCE_SCHEMA,
SOURCE_VERSION,
EXPLAIN_LEVEL,
STMTNO,
SECTNO)
REFERENCES EXPLAIN_STATEMENT
ON DELETE CASCADE)

```

```

IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_OPERATOR
--
CREATE TABLE EXPLAIN_OPERATOR
( EXPLAIN_REQUESTER VARCHAR(128) NOT NULL,
  EXPLAIN_TIME    TIMESTAMP NOT
NULL,
  SOURCE_NAME     VARCHAR(128) NOT
NULL,
  SOURCE_SCHEMA   VARCHAR(128)
NOT NULL,
  SOURCE_VERSION  VARCHAR(64)
NOT NULL,
  EXPLAIN_LEVEL   CHAR(1) NOT
NULL,
  STMTNO          INTEGER NOT NULL,
  SECTNO          INTEGER NOT NULL,
  OPERATOR_ID     INTEGER NOT
NULL,
  OPERATOR_TYPE   CHAR(6) NOT
NULL,
  TOTAL_COST      DOUBLE NOT
NULL,
  IO_COST         DOUBLE NOT NULL,
  CPU_COST        DOUBLE NOT NULL,
  FIRST_ROW_COST  DOUBLE NOT
NULL,
  RE_TOTAL_COST   DOUBLE NOT
NULL,
  RE_IO_COST      DOUBLE NOT
NULL,
  RE_CPU_COST     DOUBLE NOT
NULL,
  COMM_COST       DOUBLE NOT
NULL,
  FIRST_COMM_COST DOUBLE NOT
NULL,
  BUFFERS         DOUBLE NOT NULL,
  REMOTE_TOTAL_COST DOUBLE
NOT NULL,
  REMOTE_COMM_COST DOUBLE
NOT NULL,
  FOREIGN KEY (EXPLAIN_REQUESTER,
    EXPLAIN_TIME,
    SOURCE_NAME,
    SOURCE_SCHEMA,
    SOURCE_VERSION,
    EXPLAIN_LEVEL,
    STMTNO,
    SECTNO)
REFERENCES EXPLAIN_STATEMENT
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_PREDICATE
--
CREATE TABLE EXPLAIN_PREDICATE
( EXPLAIN_REQUESTER VARCHAR(128) NOT NULL,

```

```

NULL,
  EXPLAIN_TIME    TIMESTAMP NOT
NULL,
  SOURCE_NAME     VARCHAR(128) NOT
NULL,
  SOURCE_SCHEMA   VARCHAR(128)
NOT NULL,
  SOURCE_VERSION  VARCHAR(64)
NOT NULL,
  EXPLAIN_LEVEL   CHAR(1) NOT
NULL,
  STMTNO          INTEGER NOT NULL,
  SECTNO          INTEGER NOT NULL,
  OPERATOR_ID     INTEGER NOT
NULL,
  PREDICATE_ID    INTEGER NOT
NULL,
  HOW_APPLIED     CHAR(5) NOT
NULL,
  WHEN_EVALUATED  CHAR(3) NOT
NULL,
  RELOP_TYPE      CHAR(2) NOT
NULL,
  SUBQUERY        CHAR(1) NOT
NULL,
  FILTER_FACTOR   DOUBLE NOT
NULL,
  PREDICATE_TEXT  CLOB(2M) NOT
LOGGED,
  FOREIGN KEY (EXPLAIN_REQUESTER,
    EXPLAIN_TIME,
    SOURCE_NAME,
    SOURCE_SCHEMA,
    SOURCE_VERSION,
    EXPLAIN_LEVEL,
    STMTNO,
    SECTNO)
REFERENCES EXPLAIN_STATEMENT
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_STREAM
--
CREATE TABLE EXPLAIN_STREAM
( EXPLAIN_REQUESTER VARCHAR(128) NOT NULL,
  EXPLAIN_TIME    TIMESTAMP NOT
NULL,
  SOURCE_NAME     VARCHAR(128) NOT
NULL,
  SOURCE_SCHEMA   VARCHAR(128) NOT
NULL,
  SOURCE_VERSION  VARCHAR(64) NOT
NULL,
  EXPLAIN_LEVEL   CHAR(1) NOT
NULL,
  STMTNO          INTEGER NOT NULL,
  SECTNO          INTEGER NOT NULL,
  STREAM_ID       INTEGER NOT NULL,
  SOURCE_TYPE     CHAR(1) NOT
NULL,
  SOURCE_ID       INTEGER NOT NULL,
  TARGET_TYPE     CHAR(1) NOT NULL,

```

```

  TARGET_ID       INTEGER NOT NULL,
  OBJECT_SCHEMA  VARCHAR(128),
  OBJECT_NAME    VARCHAR(128),
  STREAM_COUNT   DOUBLE NOT
NULL,
  COLUMN_COUNT   SMALLINT NOT
NULL,
  PREDICATE_ID   INTEGER NOT
NULL,
  COLUMN_NAMES   CLOB(2M) NOT
LOGGED,
  PMID           SMALLINT NOT NULL,
  SINGLE_NODE    CHAR(5),
  PARTITION_COLUMNS CLOB(2M) NOT
LOGGED,
  FOREIGN KEY (EXPLAIN_REQUESTER,
    EXPLAIN_TIME,
    SOURCE_NAME,
    SOURCE_SCHEMA,
    SOURCE_VERSION,
    EXPLAIN_LEVEL,
    STMTNO,
    SECTNO)
REFERENCES EXPLAIN_STATEMENT
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_DIAGNOSTIC
--
CREATE TABLE EXPLAIN_DIAGNOSTIC
( EXPLAIN_REQUESTER VARCHAR(128) NOT NULL,
  EXPLAIN_TIME    TIMESTAMP NOT
NULL,
  SOURCE_NAME     VARCHAR(128)
NOT NULL,
  SOURCE_SCHEMA   VARCHAR(128)
NOT NULL,
  SOURCE_VERSION  VARCHAR(64)
NOT NULL,
  EXPLAIN_LEVEL   CHAR(1) NOT
NULL,
  STMTNO          INTEGER NOT NULL,
  SECTNO          INTEGER NOT NULL,
  DIAGNOSTIC_ID   INTEGER NOT
NULL,
  CODE            INTEGER NOT NULL,
  PRIMARY KEY (EXPLAIN_REQUESTER,
    EXPLAIN_TIME,
    SOURCE_NAME,
    SOURCE_SCHEMA,
    SOURCE_VERSION,
    EXPLAIN_LEVEL,
    STMTNO,
    SECTNO,
    DIAGNOSTIC_ID),
  FOREIGN KEY (EXPLAIN_REQUESTER,
    EXPLAIN_TIME,
    SOURCE_NAME,
    SOURCE_SCHEMA,
    SOURCE_VERSION,

```

```

        EXPLAIN_LEVEL,
        STMTNO,
        SECTNO)
REFERENCES EXPLAIN_STATEMENT
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;

--
-- EXPLAIN_DIAGNOSTIC_TOKEN
--
CREATE TABLE EXPLAIN_DIAGNOSTIC_DATA
( EXPLAIN_REQUESTER VARCHAR(128) NOT NULL,
  EXPLAIN_TIME    TIMESTAMP
  NOT NULL,
  SOURCE_NAME     VARCHAR(128)
  NOT NULL,
  SOURCE_SCHEMA   VARCHAR(128) NOT NULL,
  SOURCE_VERSION  VARCHAR(64)
  NOT NULL,
  EXPLAIN_LEVEL  CHAR(1) NOT
  NULL,
  STMTNO         INTEGER NOT
  NULL,
  SECTNO         INTEGER NOT
  NULL,
  DIAGNOSTIC_ID  INTEGER NOT
  NULL,
  ORDINAL        INTEGER NOT
  NULL,
  TOKEN          VARCHAR(1000),
  TOKEN_LONG     BLOB(3M) NOT
  LOGGED,
  FOREIGN KEY
(EXPLAIN_REQUESTER,
  EXPLAIN_TIME,
  SOURCE_NAME,
  SOURCE_SCHEMA,
  SOURCE_VERSION,
  EXPLAIN_LEVEL,
  STMTNO,
  SECTNO,
  DIAGNOSTIC_ID)
REFERENCES
EXPLAIN_DIAGNOSTIC
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;

--
-- ADVISE TABLES
--
--
-- ADVISE_INSTANCE
--
-- (must be defined first due to referential integrity definitions)
--
CREATE TABLE ADVISE_INSTANCE (
  START_TIME    TIMESTAMP NOT NULL WITH
  DEFAULT CURRENT TIMESTAMP,

```

```

  END_TIME      TIMESTAMP NOT NULL WITH
  DEFAULT CURRENT TIMESTAMP,
  MODE          VARCHAR(4) NOT NULL WITH
  DEFAULT "",
  WKLD_COMPRESSION CHAR(4) NOT NULL WITH
  DEFAULT 'NONE',
  STATUS        CHAR(9) NOT NULL WITH DEFAULT
  "",
  PRIMARY KEY (START_TIME))
IN USERSPACE1
INDEX IN USERSPACE1;

--
-- ADVISE_INDEX
--
CREATE TABLE ADVISE_INDEX(
  EXPLAIN_REQUESTER VARCHAR(128) NOT NULL
  WITH DEFAULT "",
  EXPLAIN_TIME    TIMESTAMP NOT NULL WITH
  DEFAULT CURRENT TIMESTAMP,
  SOURCE_NAME     VARCHAR(128) NOT NULL WITH
  DEFAULT "",
  SOURCE_SCHEMA   VARCHAR(128) NOT NULL WITH
  DEFAULT "",
  SOURCE_VERSION  VARCHAR(64) NOT NULL WITH
  DEFAULT "",
  EXPLAIN_LEVEL  CHAR(1) NOT NULL WITH
  DEFAULT "",
  STMTNO         INTEGER NOT NULL WITH
  DEFAULT 0,
  SECTNO         INTEGER NOT NULL WITH
  DEFAULT 0,
  QUERYNO        INTEGER NOT NULL WITH
  DEFAULT 0,
  QUERYTAG       CHAR(20) NOT NULL WITH
  DEFAULT "",
  NAME           VARCHAR(128) NOT NULL,
  CREATOR        VARCHAR(128) NOT NULL WITH
  DEFAULT "",
  TBNAME         VARCHAR(128) NOT NULL,
  TBCREATOR      VARCHAR(128) NOT NULL WITH
  DEFAULT "",
  COLNAMES       CLOB(2M) NOT NULL,
  UNIQUERULE     CHAR(1) NOT NULL WITH
  DEFAULT "",
  COLCOUNT      SMALLINT NOT NULL WITH
  DEFAULT 0,
  IID            SMALLINT NOT NULL WITH DEFAULT 0,
  NLEAF          INTEGER NOT NULL WITH DEFAULT
  0,
  NLEVELS        SMALLINT NOT NULL WITH
  DEFAULT 0,
  FIRSTKEYCARD   BIGINT NOT NULL WITH
  DEFAULT 0,
  FULLKEYCARD    BIGINT NOT NULL WITH
  DEFAULT 0,
  CLUSTERRATIO   SMALLINT NOT NULL WITH
  DEFAULT 0,
  CLUSTERFACTOR  DOUBLE NOT NULL WITH
  DEFAULT 0,

```

```

  USERDEFINED    SMALLINT NOT NULL WITH
  DEFAULT 0,
  SYSTEM_REQUIRED SMALLINT NOT NULL WITH
  DEFAULT 0,
  CREATE_TIME    TIMESTAMP NOT NULL WITH
  DEFAULT CURRENT TIMESTAMP,
  STATS_TIME     TIMESTAMP WITH DEFAULT
  CURRENT TIMESTAMP,
  PAGE_FETCH_PAIRS VARCHAR(254) NOT NULL
  WITH DEFAULT "",
  REMARKS        VARCHAR(254) WITH DEFAULT
  "",
  DEFINER        VARCHAR(128) NOT NULL WITH
  DEFAULT "",
  CONVERTED      CHAR(1) NOT NULL WITH
  DEFAULT "",
  SEQUENTIAL_PAGES INTEGER NOT NULL WITH
  DEFAULT 0,
  DENSITY        INTEGER NOT NULL WITH
  DEFAULT 0,
  FIRST2KEYCARD  BIGINT NOT NULL WITH
  DEFAULT 0,
  FIRST3KEYCARD  BIGINT NOT NULL WITH
  DEFAULT 0,
  FIRST4KEYCARD  BIGINT NOT NULL WITH
  DEFAULT 0,
  PCTFREE        SMALLINT NOT NULL WITH
  DEFAULT -1,
  UNIQUE_COLCOUNT SMALLINT NOT NULL WITH
  DEFAULT -1,
  MINPCTUSED     SMALLINT NOT NULL WITH
  DEFAULT 0,
  REVERSE_SCANS  CHAR(1) NOT NULL WITH
  DEFAULT 'N',
  USE_INDEX       CHAR(1),
  CREATION_TEXT  CLOB(2M) NOT NULL NOT
  LOGGED WITH DEFAULT "",
  PACKED_DESC    BLOB(1M) NOT LOGGED,
  RUN_ID         TIMESTAMP,
  INDEXTYPE      VARCHAR(4) NOT NULL WITH
  DEFAULT "",
  EXISTS         CHAR(1) NOT NULL WITH DEFAULT
  'N',
  RIDTOBLOCK     CHAR(1) NOT NULL WITH
  DEFAULT 'N',
  FOREIGN KEY (RUN_ID)
REFERENCES ADVISE_INSTANCE
(START_TIME)
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;

--
-- ADVISE_WORKLOAD
--
CREATE TABLE ADVISE_WORKLOAD (
  WORKLOAD_NAME CHAR(128) NOT NULL WITH
  DEFAULT 'WK0',
  STATEMENT_NO  INTEGER NOT NULL WITH
  DEFAULT 1,
  STATEMENT_TEXT CLOB(2M) NOT NULL NOT
  LOGGED,

```

```

STATEMENT_TAG VARCHAR(256) NOT NULL WITH
DEFAULT "",
FREQUENCY INTEGER NOT NULL WITH
DEFAULT 1,
IMPORTANCE DOUBLE NOT NULL WITH
DEFAULT 1,
WEIGHT DOUBLE NOT NULL WITH DEFAULT
1,
COST_BEFORE DOUBLE,
COST_AFTER DOUBLE,
COMPILABLE CHAR(17))
IN USERSPACE1
INDEX IN USERSPACE1;

--
-- ADVISE_MQT
--
CREATE TABLE ADVISE_MQT (
EXPLAIN_REQUESTER VARCHAR(128) NOT NULL
WITH DEFAULT "",
EXPLAIN_TIME TIMESTAMP NOT NULL WITH
DEFAULT CURRENT_TIMESTAMP,
SOURCE_NAME VARCHAR(128) NOT NULL WITH
DEFAULT "",
SOURCE_SCHEMA VARCHAR(128) NOT NULL WITH
DEFAULT "",
SOURCE_VERSION VARCHAR(64) NOT NULL WITH
DEFAULT "",
EXPLAIN_LEVEL CHAR(1) NOT NULL WITH
DEFAULT "",
STMTNO INTEGER NOT NULL WITH
DEFAULT 0,
SECTNO INTEGER NOT NULL WITH
DEFAULT 0,
NAME VARCHAR(128) NOT NULL,
CREATOR VARCHAR(128) NOT NULL WITH
DEFAULT "",
IID SMALLINT NOT NULL WITH DEFAULT 0,
CREATE_TIME TIMESTAMP NOT NULL WITH
DEFAULT CURRENT_TIMESTAMP,
STATS_TIME TIMESTAMP WITH DEFAULT
CURRENT_TIMESTAMP,
NUMROWS DOUBLE NOT NULL WITH
DEFAULT 0,
NUMCOLS SMALLINT NOT NULL WITH
DEFAULT 0,
ROWSIZE DOUBLE NOT NULL WITH
DEFAULT 0,
BENEFIT FLOAT NOT NULL WITH DEFAULT
0.0,
USE_MQT CHAR(1),
MQT_SOURCE CHAR(1),
QUERY_TEXT CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT "",
CREATION_TEXT CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT "",
SAMPLE_TEXT CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT "",
COLSTATS CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT "",
EXTRA_INFO BLOB(2M) NOT NULL NOT
LOGGED with default BLOB("),

```

```

TBSpace VARCHAR(128) NOT NULL WITH
DEFAULT "",
RUN_ID TIMESTAMP,
REFRESH_TYPE CHAR(1) NOT NULL WITH
DEFAULT "",
EXISTS CHAR(1) NOT NULL WITH DEFAULT
'N',
FOREIGN KEY (RUN_ID)
REFERENCES ADVISE_INSTANCE
(START_TIME)
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;

--
-- ADVISE_PARTITION
--
CREATE TABLE ADVISE_PARTITION (
EXPLAIN_REQUESTER VARCHAR(128) NOT NULL
WITH DEFAULT "",
EXPLAIN_TIME TIMESTAMP NOT NULL WITH
DEFAULT CURRENT_TIMESTAMP,
SOURCE_NAME VARCHAR(128) NOT NULL WITH
DEFAULT "",
SOURCE_SCHEMA VARCHAR(128) NOT NULL WITH
DEFAULT "",
SOURCE_VERSION VARCHAR(64) NOT NULL WITH
DEFAULT "",
EXPLAIN_LEVEL CHAR(1) NOT NULL WITH
DEFAULT "",
STMTNO INTEGER NOT NULL WITH
DEFAULT 0,
SECTNO INTEGER NOT NULL WITH
DEFAULT 0,
QUERYNO INTEGER NOT NULL WITH
DEFAULT 0,
QUERYTAG CHAR(20) NOT NULL WITH
DEFAULT "",
TBNAME VARCHAR(128) NOT NULL,
TBCREATOR VARCHAR(128) NOT NULL WITH
DEFAULT "",
PMID SMALLINT NOT NULL,
TBSpace VARCHAR(128) NOT NULL WITH
DEFAULT "",
COLNAMES CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT "",
COLCOUNT SMALLINT NOT NULL WITH
DEFAULT 0,
REPLICATE CHAR(1) NOT NULL WITH
DEFAULT 'N',
COST DOUBLE NOT NULL,
USEIT CHAR(1),
RUN_ID TIMESTAMP,
FOREIGN KEY (RUN_ID)
REFERENCES ADVISE_INSTANCE
(START_TIME)
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;

--
-- ADVISE_TABLE

```

```

--
CREATE TABLE ADVISE_TABLE (
RUN_ID TIMESTAMP,
TABLE_NAME VARCHAR(128) NOT NULL,
TABLE_SCHEMA VARCHAR(128) NOT NULL WITH
DEFAULT "",
TABLESPACE VARCHAR(128) NOT NULL WITH
DEFAULT "",
SELECTION_FLAG VARCHAR(8) NOT NULL WITH
DEFAULT "",
TABLE_EXISTS CHAR(1) NOT NULL WITH
DEFAULT "",
USE_TABLE CHAR(1) NOT NULL WITH
DEFAULT "",
GEN_COLUMNS CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT "",
ORGANIZE_BY CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT "",
CREATION_TEXT CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT "",
ALTER_COMMAND CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT "",
DISKUSE DOUBLE NOT NULL WITH
DEFAULT 0,
FOREIGN KEY (RUN_ID)
REFERENCES ADVISE_INSTANCE
(START_TIME)
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;

--
-- Commit work
--
COMMIT WORK;

--
-- Optional Indexes: The following indexes are recommended
for improved performance
-- of explain-related utilities. These create index statements can
be deleted, or
-- the indexes dropped if space is a problem.
--
CREATE INDEX STMT_I1 on
EXPLAIN_STATEMENT(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO);
CREATE INDEX ARG_I1 on
EXPLAIN_ARGUMENT(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO, OPERATOR_ID);
CREATE INDEX PRD_I1 on
EXPLAIN_PREDICATE(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO, OPERATOR_ID);
CREATE INDEX OPR_I1 on
EXPLAIN_OPERATOR(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO, OPERATOR_ID);
CREATE INDEX STM_I1 on
EXPLAIN_STREAM(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO);
CREATE INDEX OBJ_I1 on
EXPLAIN_OBJECT(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO);

```

```
CREATE INDEX EXP_DIAG_DAT_I1 on
  EXPLAIN_DIAGNOSTIC_DATA(EXPLAIN_TIME,
  EXPLAIN_LEVEL, STMTNO, SECTNO, DIAGNOSTIC_ID,
  ORDINAL);
```

```
CREATE INDEX IDX_I1 on
  ADVISE_INDEX (EXPLAIN_TIME);
CREATE INDEX IDX_I2 on
  ADVISE_INDEX (NAME, EXPLAIN_TIME);
CREATE INDEX MQT_I1 on
  ADVISE_MQT (EXPLAIN_TIME);
CREATE INDEX MQT_I2 on
  ADVISE_MQT (NAME, EXPLAIN_TIME);
CREATE INDEX PRT_I1 on
  ADVISE_PARTITION (EXPLAIN_TIME);
```

```
--
-- Commit work
--
COMMIT WORK;
```

```
--
-- The following function formats the explain diagnostic table
messages
```

```
CREATE FUNCTION
  EXPLAIN_GET_MSGS( EXPLAIN_REQUESTER
  VARCHAR(128),
  EXPLAIN_TIME   TIMESTAMP,
  SOURCE_NAME    VARCHAR(128),
  SOURCE_SCHEMA  VARCHAR(128),
  SOURCE_VERSION VARCHAR(64),
  EXPLAIN_LEVEL  CHAR(1),
  STMTNO        INTEGER,
  SECTNO        INTEGER,
  LOCALE        VARCHAR(33) )
```

```
RETURNS TABLE ( EXPLAIN_REQUESTER
  VARCHAR(128),
  EXPLAIN_TIME   TIMESTAMP,
  SOURCE_NAME    VARCHAR(128),
  SOURCE_SCHEMA  VARCHAR(128),
  SOURCE_VERSION VARCHAR(64),
  EXPLAIN_LEVEL  CHAR(1),
  STMTNO        INTEGER,
  SECTNO        INTEGER,
  DIAGNOSTIC_ID INTEGER,
  LOCALE        VARCHAR(33),
  MSG           VARCHAR(4096) )
```

```
SPECIFIC EXPLAIN_GET_MSGS
LANGUAGE SQL
DETERMINISTIC
NO EXTERNAL ACTION
READS SQL DATA
RETURN SELECT A.A_EXPLAIN_REQUESTER,
  A.A_EXPLAIN_TIME,
  A.A_SOURCE_NAME,
  A.A_SOURCE_SCHEMA,
  A.A_SOURCE_VERSION,
  A.A_EXPLAIN_LEVEL,
  A.A_STMTNO,
  A.A_SECTNO,
  A.A_DIAGNOSTIC_ID,
```

```
  F.LOCALE,
  F.MSG
FROM EXPLAIN_DIAGNOSTIC
A( A_EXPLAIN_REQUESTER,
  A_EXPLAIN_TIME,
  A_SOURCE_NAME,
  A_SOURCE_SCHEMA,
  A_SOURCE_VERSION,
  A_EXPLAIN_LEVEL,
  A_STMTNO,
  A_SECTNO,
  A_DIAGNOSTIC_ID,
  A_CODE ),
TABLE( SYSPROC.EXPLAIN_GET_MSGS(
  CAST( NULL AS VARCHAR(33) ),
  A.A_CODE,
  ( SELECT TOKEN FROM
  EXPLAIN_DIAGNOSTIC_DATA B
  WHERE A.A_EXPLAIN_REQUESTER =
  B.EXPLAIN_REQUESTER
  AND A.A_EXPLAIN_TIME =
  B.EXPLAIN_TIME
  AND A.A_SOURCE_NAME =
  B.SOURCE_NAME
  AND A.A_SOURCE_SCHEMA =
  B.SOURCE_SCHEMA
  AND A.A_SOURCE_VERSION =
  B.SOURCE_VERSION
  AND A.A_EXPLAIN_LEVEL =
  B.EXPLAIN_LEVEL
  AND A.A_STMTNO = B.STMTNO
  AND A.A_SECTNO = B.SECTNO
  AND A.A_DIAGNOSTIC_ID =
  B.DIAGNOSTIC_ID
  AND B.ORDINAL=1 ),
  ( SELECT TOKEN FROM
  EXPLAIN_DIAGNOSTIC_DATA B
  WHERE A.A_EXPLAIN_REQUESTER =
  B.EXPLAIN_REQUESTER
  AND A.A_EXPLAIN_TIME =
  B.EXPLAIN_TIME
  AND A.A_SOURCE_NAME =
  B.SOURCE_NAME
  AND A.A_SOURCE_SCHEMA =
  B.SOURCE_SCHEMA
  AND A.A_SOURCE_VERSION =
  B.SOURCE_VERSION
  AND A.A_EXPLAIN_LEVEL =
  B.EXPLAIN_LEVEL
  AND A.A_STMTNO = B.STMTNO
  AND A.A_SECTNO = B.SECTNO
  AND A.A_DIAGNOSTIC_ID =
  B.DIAGNOSTIC_ID
  AND B.ORDINAL=2 ),
  ( SELECT TOKEN FROM
  EXPLAIN_DIAGNOSTIC_DATA B
  WHERE A.A_EXPLAIN_REQUESTER =
  B.EXPLAIN_REQUESTER
  AND A.A_EXPLAIN_TIME =
  B.EXPLAIN_TIME
  AND A.A_SOURCE_NAME =
  B.SOURCE_NAME
  AND A.A_SOURCE_SCHEMA =
  B.SOURCE_SCHEMA
  AND A.A_SOURCE_VERSION =
  B.SOURCE_VERSION
  AND A.A_EXPLAIN_LEVEL =
  B.EXPLAIN_LEVEL
  AND A.A_STMTNO = B.STMTNO
  AND A.A_SECTNO = B.SECTNO
  AND A.A_DIAGNOSTIC_ID =
  B.DIAGNOSTIC_ID
  AND B.ORDINAL=2 ),
  ( SELECT TOKEN FROM
  EXPLAIN_DIAGNOSTIC_DATA B
  WHERE A.A_EXPLAIN_REQUESTER =
  B.EXPLAIN_REQUESTER
  AND A.A_EXPLAIN_TIME =
  B.EXPLAIN_TIME
  AND A.A_SOURCE_NAME =
  B.SOURCE_NAME
```

```
AND A.A_SOURCE_SCHEMA =
  B.SOURCE_SCHEMA
  AND A.A_SOURCE_VERSION =
  B.SOURCE_VERSION
  AND A.A_EXPLAIN_LEVEL =
  B.EXPLAIN_LEVEL
  AND A.A_STMTNO = B.STMTNO
  AND A.A_SECTNO = B.SECTNO
  AND A.A_DIAGNOSTIC_ID =
  B.DIAGNOSTIC_ID
  AND B.ORDINAL=3 ) ) F
WHERE ( EXPLAIN_REQUESTER IS NULL OR
  EXPLAIN_REQUESTER =
  A.A_EXPLAIN_REQUESTER )
AND ( EXPLAIN_TIME IS NULL OR
  EXPLAIN_TIME = A.A_EXPLAIN_TIME )
AND ( SOURCE_NAME IS NULL OR
  SOURCE_NAME = A.A_SOURCE_NAME )
AND ( SOURCE_SCHEMA IS NULL OR
  SOURCE_SCHEMA =
  A.A_SOURCE_SCHEMA )
AND ( SOURCE_VERSION IS NULL OR
  SOURCE_VERSION =
  A.A_SOURCE_VERSION )
AND ( EXPLAIN_LEVEL IS NULL OR
  EXPLAIN_LEVEL = A.A_EXPLAIN_LEVEL )
AND ( STMTNO IS NULL OR
  STMTNO = A.A_STMTNO )
AND ( SECTNO IS NULL OR
  SECTNO = A.A_SECTNO );
```

```
--
-- Commit work
--
COMMIT WORK;
```

utils/UNEXPLAIN.DDL

```
-----
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996
-- 2006
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with
  IBM Corp.
```

```
-----
DROP INDEX STMT_I1;
DROP INDEX ARG_I1;
DROP INDEX PRD_I1;
DROP INDEX OPR_I1;
DROP INDEX STM_I1;
DROP INDEX OBJ_I1;
DROP TABLE EXPLAIN_INSTANCE;
DROP TABLE EXPLAIN_STATEMENT;
```



```
DROP TABLE EXPLAIN_ARGUMENT;  
DROP TABLE EXPLAIN_OBJECT;  
DROP TABLE EXPLAIN_OPERATOR;  
DROP TABLE EXPLAIN_PREDICATE;  
DROP TABLE EXPLAIN_STREAM;  
DROP TABLE ADVISE_INDEX;  
DROP TABLE ADVISE_WORKLOAD;
```

Appendix - B: Tunable Parameters

B.1 Database Parameters.

db.cfg.out

Database Configuration for Database TPCC

Database configuration release level = 0x0a00
 Database release level = 0x0a00

Database territory = US
 Database code page = 819
 Database code set = ISO8859-1
 Database country/region code = 1
 Database collating sequence = BINARY
 Alternate collating sequence (ALT_COLLATE) =
 Database page size = 4096

Dynamic SQL Query management (DYN_QUERY_MGMT) = DISABLE

Discovery support for this database (DISCOVER_DB) = ENABLE

Default query optimization class (DFT_QUERYOPT) = 5
 Degree of parallelism (DFT_DEGREE) = 1
 Continue upon arithmetic exceptions (DFT_SQLMATHWARN) = NO
 Default refresh age (DFT_REFRESH_AGE) = 0
 Default maintained table types for opt (DFT_MTTB_TYPES) = SYSTEM
 Number of frequent values retained (NUM_FREQVALUES) = 10
 Number of quantiles retained (NUM_QUANTILES) = 20

Backup pending = NO

Database is consistent = NO
 Rollforward pending = NO
 Restore pending = NO

Multi-page file allocation enabled = YES

Log retain for recovery status = RECOVERY
 User exit for logging status = NO

Data Links Token Expiry Interval (sec) (DL_EXPINT) = 60
 Data Links Write Token Init Expiry Intvl(DL_WT_IEXPINT) = 60
 Data Links Number of Copies (DL_NUM_COPIES) = 1
 Data Links Time after Drop (days) (DL_TIME_DROP) = 1
 Data Links Token in Uppercase (DL_UPPER) = NO
 Data Links Token Algorithm (DL_TOKEN) = MACO

Database heap (4KB) (DBHEAP) = 524288

Size of database shared memory (4KB) (DATABASE_MEMORY) = 128779200
 Catalog cache size (4KB) (CATALOGCACHE_SZ) = (MAXAPPLS*4)
 Log buffer size (4KB) (LOGBUFSZ) = 60000
 Utilities heap size (4KB) (UTIL_HEAP_SZ) = 5000
 Buffer pool size (pages) (BUFFPAGE) = 1000
 Extended storage segments size (4KB) (ESTORE_SEG_SZ) = 16000
 Number of extended storage segments (NUM_ESTORE_SEGS) = 0
 Max storage for lock list (4KB) (LOCKLIST) = 40000

Max size of appl. group mem set (4KB) (APPGROUP_MEM_SZ) = 20000
 Percent of mem for appl. group heap (GROUPHEAP_RATIO) = 70
 Max appl. control heap size (4KB) (APP_CTL_HEAP_SZ) = 128

Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = (SHEAPTHRES)
 Sort list heap (4KB) (SORTHEAP) = 16
 SQL statement heap (4KB) (STMTHEAP) = 65000
 Default application heap (4KB) (APPLHEAPSZ) = 2500
 Package cache size (4KB) (PCKCACHESZ) = 50000
 Statistics heap size (4KB) (STAT_HEAP_SZ) = 4384

Interval for checking deadlock (ms) (DLCHKTIME) = 3000
 Percent. of lock lists per application (MAXLOCKS) = 20
 Lock timeout (sec) (LOCKTIMEOUT) = -1

Changed pages threshold (CHNGPGS_THRESH) = 99
 Number of asynchronous page cleaners (NUM_IOCLEANERS) = 20
 Number of I/O servers (NUM_IOSERVERS) = 1
 Index sort flag (INDEXSORT) = YES
 Sequential detect flag (SEQDETECT) = NO
 Default prefetch size (pages) (DFT_PREFETCH_SZ) = AUTOMATIC

Track modified pages (TRACKMOD) = OFF

Default number of containers = 1
 Default tablespace extentsize (pages) (DFT_EXTENT_SZ) = 32

Max number of active applications (MAXAPPLS) = 5050
 Average number of active applications (AVG_APPLS) = 1
 Max DB files open per application (MAXFILOP) = 800

Log file size (4KB) (LOGFILSIZ) = 262144
 Number of primary log files (LOGPRIMARY) = 250
 Number of secondary log files (LOGSECOND) = 0
 Changed path to log files (NEWLOGPATH) =
 Path to log files = /dev/rdbloglv
 Overflow log path (OVERFLOWLOGPATH) =
 Mirror log path (MIRRORLOGPATH) =
 First active log file = S0000001.LOG
 Block log on disk full (BLK_LOG_DSK_FUL) = NO

Percent of max active log space by transaction(MAX_LOG) = 0
 Num. of active log files for 1 active UOW(NUM_LOG_SPAN) = 0

Group commit count (MINCOMMIT) = 3
 Percent log file reclaimed before soft ckcpt (SOFTMAX) = 6587
 Log retain for recovery enabled (LOGRETAIN) = RECOVERY
 User exit for logging enabled (USEREXIT) = OFF

HADR database role = STANDARD
 HADR local host name (HADR_LOCAL_HOST) =
 HADR local service name (HADR_LOCAL_SVC) =
 HADR remote host name (HADR_REMOTE_HOST) =
 HADR remote service name (HADR_REMOTE_SVC) =
 HADR instance name of remote server (HADR_REMOTE_INST) =
 HADR timeout value (HADR_TIMEOUT) = 120
 HADR log write synchronization mode (HADR_SYNCMODE) = NEARSYNC

First log archive method (LOGARCHMETH1) = LOGRETAIN
 Options for logarchmeth1 (LOGARCHOPT1) =
 Second log archive method (LOGARCHMETH2) = OFF
 Options for logarchmeth2 (LOGARCHOPT2) =
 Failover log archive path (FAILARCHPATH) =
 Number of log archive retries on error (NUMARCHRETRY) = 5
 Log archive retry Delay (secs) (ARCHRETRYDELAY) = 20
 Vendor options (VENDOROPT) =

Auto restart enabled (AUTORESTART) = ON
 Index re-creation time and redo index build (INDEXREC) = SYSTEM (RESTART)
 Log pages during index build (LOGINDEXBUILD) = OFF
 Default number of loadrec sessions (DFT_LOADREC_SES) = 1
 Number of database backups to retain (NUM_DB_BACKUPS) = 12
 Recovery history retention (days) (REC_HIS_RETENTN) = 366

TSM management class (TSM_MGMTCLASS) =
 TSM node name (TSM_NODENAME) =
 TSM owner (TSM_OWNER) =
 TSM password (TSM_PASSWORD) =

Automatic maintenance (AUTO_MAINT) = OFF
 Automatic database backup (AUTO_DB_BACKUP) = OFF
 Automatic table maintenance (AUTO_TBL_MAINT) = OFF
 Automatic runstats (AUTO_RUNSTATS) = OFF
 Automatic statistics profiling (AUTO_STATS_PROF) = OFF
 Automatic profile updates (AUTO_PROF_UPD) = OFF
 Automatic reorganization (AUTO_REORG) = OFF

dbm.cfg.out

Database Manager Configuration

Node type = Database Server with local clients

Database manager configuration release level = 0x0a00

CPU speed (millisec/instruction) (CPUSPEED) =
3.148961e-07

Max number of concurrently active databases (NUMDB) = 1
Data Links support (DALINKS) = NO
Federated Database System Support (FEDERATED) =
NO

Transaction processor monitor name (TP_MON_NAME) =

Default charge-back account (DFT_ACCOUNT_STR) =

Java Development Kit installation path (JDK_PATH) =
/usr/java14_64

Diagnostic error capture level (DIAGLEVEL) = 1
Notify Level (NOTIFYLEVEL) = 1
Diagnostic data directory path (DIAGPATH) =

Default database monitor switches

Buffer pool (DFT_MON_BUFPOOL) = OFF
Lock (DFT_MON_LOCK) = OFF
Sort (DFT_MON_SORT) = OFF
Statement (DFT_MON_STMT) = OFF
Table (DFT_MON_TABLE) = OFF
Timestamp (DFT_MON_TIMESTAMP) = OFF
Unit of work (DFT_MON_UOW) = OFF
Monitor health of instance and databases (HEALTH_MON) =
OFF

SYSDM group name (SYSDM_GROUP) =
STAFF
SYSCTRL group name (SYSCTRL_GROUP) =
SYSMAINT group name (SYSMAINT_GROUP) =
SYSMON group name (SYSMON_GROUP) =

Client Userid-Password Plugin (CLNT_PW_PLUGIN) =
Client Kerberos Plugin (CLNT_KRB_PLUGIN) =
Group Plugin (GROUP_PLUGIN) =
GSS Plugin for Local Authorization (LOCAL_GSSPLUGIN) =
Server Plugin Mode (SRV_PLUGIN_MODE) =
UNFENCED

Server List of GSS Plugins (SRVCON_GSSPLUGIN_LIST)
=

Server Userid-Password Plugin (SRVCON_PW_PLUGIN)
=

Server Connection Authentication (SRVCON_AUTH) =
NOT_SPECIFIED

Database manager authentication (AUTHENTICATION) =
CLIENT

Cataloging allowed without authority (CATALOG_NOAUTH) =
YES

Trust all clients (TRUST_ALLCLNTS) = YES

Trusted client authentication (TRUST_CLNTAUTH) =
CLIENT
Bypass federated authentication (FED_NOAUTH) = NO

Default database path (DFTDBPATH) =
/home/tpcc

Database monitor heap size (4KB) (MON_HEAP_SZ) =
4096

Java Virtual Machine heap size (4KB) (JAVA_HEAP_SZ) =
1024

Audit buffer size (4KB) (AUDIT_BUF_SZ) = 0

Size of instance shared memory (4KB)
(INSTANCE_MEMORY) = AUTOMATIC

Backup buffer default size (4KB) (BACKBUFSZ) = 1024

Restore buffer default size (4KB) (RESTBUFSZ) = 1024

Sort heap threshold (4KB) (SHEAPTHRES) = 20000

Directory cache support (DIR_CACHE) = YES

Application support layer heap size (4KB) (ASLHEAPSZ) = 15
Max requester I/O block size (bytes) (RQRIOBLK) = 4096
Query heap size (4KB) (QUERY_HEAP_SZ) = 1000

Workload impact by throttled utilities(UTIL_IMPACT_LIM) = 10

Priority of agents (AGENTPRI) = 60
Max number of existing agents (MAXAGENTS) = 5050
Agent pool size (NUM_POOLAGENTS) = 700
Initial number of agents in pool (NUM_INITAGENTS) = 0
Max number of coordinating agents
(MAX_COORDAGENTS) = MAXAGENTS
Max no. of concurrent coordinating agents (MAXCAGENTS) =
MAX_COORDAGENTS
Max number of client connections (MAX_CONNECTIONS)
= MAX_COORDAGENTS

Keep fenced process (KEEPFENCED) = YES
Number of pooled fenced processes (FENCED_POOL) =
MAX_COORDAGENTS
Initial number of fenced processes (NUM_INITFENCED) = 0

Index re-creation time and redo index build (INDEXREC) =
RESTART

Transaction manager database name (TM_DATABASE) =
1ST_CONN

Transaction resync interval (sec) (RESYNC_INTERVAL) =
180

SPM name (SPM_NAME) =
SPM log size (SPM_LOG_FILE_SZ) = 256
SPM resync agent limit (SPM_MAX_RESYNC) = 20
SPM log path (SPM_LOG_PATH) =

TCP/IP Service name (SVCENAME) =

db2ctpc
Discovery mode (DISCOVER) = SEARCH
Discover server instance (DISCOVER_INST) =
ENABLE

Maximum query degree of parallelism
(MAX_QUERYDEGREE) = ANY
Enable intra-partition parallelism (INTRA_PARALLEL) = NO

No. of int. communication buffers(4KB)(FCM_NUM_BUFFERS)
= 512

Number of FCM request blocks (FCM_NUM_RQB) =
AUTOMATIC

Number of FCM connection entries
(FCM_NUM_CONNECT) = AUTOMATIC

Number of FCM message anchors
(FCM_NUM_ANCHORS) = AUTOMATIC

db2set.cfg.out

[i] DB2_LARGE_PAGE_MEM=DB:16GB
[i] DB2_SELUDI_COMM_BUFFER=Y
[i] DB2_USE_ALTERNATE_PAGE_CLEANING=YES
[i] DB2_MAX_NON_TABLE_LOCKS=1000
[i] DB2_TRUSTED_BINDIN=ON
[i] DB2_KEEPTABLELOCK=ON
[i] DB2_NO_FORK_CHECK=ON
[i] DB2_APM_PERFORMANCE=ALL
[i] DB2_ENABLE_BUFDPD=OFF
[i] DB2_PINNED_BP=YES
[i] DB2_SELECTIVITY=ON
[i] DB2ASSUMEUPDATE=ON
[i] DB2CHECKCLIENTINTERVAL=0
[i] DB2_HASH_JOIN=OFF
[i] DB2CHKSQLDA=OFF
[i] DB2ENVLIST=MEMORY_AFFINITY_LDR_CNTRL
[i] DB2_COLLECT_TS_REC_INFO=false
[i] DB2COMM=tcPIP
[i] DB2CHKPTR=OFF

B.2 Transaction Monitor Parameters

tpccCom.tpcc.com.settings.txt

Transactions not supported
Enable object pooling
Minimum pool size 32
Maximum pool size 32
Creation timeout 100,000
Enable Object Construction
Enable Just in time activation
Concurrency Required

InetInfo.registry.reg

Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\InetInfo\Parameters]

"ListenBackLog"=dword:000000fa
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,00,00
"MaxConnections"=dword:000061a8
"PoolThreadLimit"=dword:00000190
"ThreadTimeout"=dword:00015180
"MaxConcurrency"=dword:ffffff

tcpip_parameters_registry.reg

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters]
"NV Hostname"="client32"
"DataBasePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6
d,00,52,00,6f,00,6f,\
00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,0
0,32,00,5c,00,\
64,00,72,00,69,00,76,00,65,00,72,00,73,00,5c,00,65,00,74,00,6
3,00,00,00
"NameServer"=""
"ForwardBroadcasts"=dword:00000000
"IPEnableRouter"=dword:00000000
"Domain"=""
"Hostname"="client32"
"SearchList"=""
"UseDomainNameDevolution"=dword:00000001
"EnableICMPRedirect"=dword:00000001
"DeadGWDetectDefault"=dword:00000001
"DontAddDefaultGatewayDefault"=dword:00000000
"EnableSecurityFilters"=dword:00000000
"AllowUnqualifiedQuery"=dword:00000000
"PrioritizeRecordData"=dword:00000001
"GlobalTcpWindowSize"=dword:00008000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters\Adapters]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters\Adapters\NdisWanlp]
"LLInterface"="WANARP"
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,6
1,00,72,00,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,6
5,00,72,00,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,36,00,31,00,44,00,33,0
0,33,00,39,00,\
37,00,37,00,2d,00,37,00,31,00,30,00,46,00,2d,00,34,00,41,00,3
7,00,45,00,2d,\
00,38,00,44,00,34,00,37,00,2d,00,34,00,42,00,34,00,38,00,32,0
0,42,00,43,00,\
35,00,32,00,33,00,46,00,38,00,7d,00,00,00,54,00,63,00,70,00,6
9,00,70,00,5c,\
9,00,70,00,5c,\
```

```
00,50,00,61,00,72,00,61,00,6d,00,65,00,74,00,65,00,72,00,73,0
0,5c,00,49,00,\
6e,00,74,00,65,00,72,00,66,00,61,00,63,00,65,00,73,00,5c,00,7
b,00,39,00,41,\
00,33,00,41,00,41,00,36,00,41,00,43,00,2d,00,35,00,38,00,34,0
0,36,00,2d,00,\
34,00,30,00,37,00,45,00,2d,00,38,00,32,00,35,00,30,00,2d,00,4
6,00,30,00,33,\
00,42,00,36,00,30,00,34,00,39,00,36,00,36,00,44,00,43,00,7d,0
0,00,00,00,00
"NumInterfaces"=dword:00000002
"IpInterfaces"=hex:77,39,d3,61,0f,71,7e,4a,8d,47,4b,48,2b,c5,2
3,f8,ac,a6,3a,9a,\
46,58,7e,40,82,50,f0,3b,60,49,66,dc
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters\Adapters\{0435C97F-9186-473F-B181-
5449A2CF0042}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,6
1,00,72,00,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,6
5,00,72,00,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,30,00,34,00,33,00,35,0
0,43,00,39,00,\
37,00,46,00,2d,00,39,00,31,00,38,00,36,00,2d,00,34,00,37,00,3
3,00,46,00,2d,\
00,42,00,31,00,38,00,31,00,2d,00,35,00,34,00,34,00,39,00,41,0
0,32,00,43,00,\
46,00,30,00,30,00,34,00,32,00,7d,00,00,00,00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters\Adapters\{1E07A95A-92A0-4836-BF73-
7AE38F8ACA07}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,6
1,00,72,00,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,6
5,00,72,00,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,31,00,45,00,30,00,37,0
0,41,00,39,00,\
35,00,41,00,2d,00,39,00,32,00,41,00,30,00,2d,00,34,00,38,00,3
3,00,36,00,2d,\
00,42,00,46,00,37,00,33,00,2d,00,37,00,41,00,45,00,33,00,38,0
0,46,00,38,00,\
41,00,43,00,41,00,30,00,37,00,7d,00,00,00,00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters\Adapters\{2EA04AA5-93A6-437F-9153-
2F6834D3B795}]
"LLInterface"=""
```

```
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,6
1,00,72,00,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,6
5,00,72,00,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,32,00,45,00,41,00,30,0
0,34,00,41,00,\
41,00,35,00,2d,00,39,00,33,00,41,00,36,00,2d,00,34,00,33,00,3
7,00,46,00,2d,\
00,39,00,31,00,35,00,33,00,2d,00,32,00,46,00,36,00,38,00,33,0
0,34,00,44,00,\
33,00,42,00,37,00,39,00,35,00,7d,00,00,00,00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters\Adapters\{37430121-7BE3-4B55-8AAB-
D8AD09B2029C}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,6
1,00,72,00,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,6
5,00,72,00,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,33,00,37,00,34,00,33,0
0,30,00,31,00,\
32,00,31,00,2d,00,37,00,42,00,45,00,33,00,2d,00,34,00,42,00,3
5,00,35,00,2d,\
00,38,00,41,00,41,00,42,00,2d,00,44,00,38,00,41,00,44,00,30,0
0,39,00,42,00,\
32,00,30,00,32,00,39,00,43,00,7d,00,00,00,00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters\Adapters\{6FE29D81-59D5-4401-A77E-
BE3BC929B6E0}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,6
1,00,72,00,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,6
5,00,72,00,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,36,00,46,00,45,00,32,0
0,39,00,44,00,\
38,00,31,00,2d,00,35,00,39,00,44,00,35,00,2d,00,34,00,34,00,3
3,00,31,00,2d,\
00,41,00,37,00,37,00,45,00,2d,00,42,00,45,00,33,00,42,00,43,0
0,39,00,32,00,\
39,00,42,00,36,00,45,00,30,00,7d,00,00,00,00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters\Adapters\{7B215199-A3F3-4836-89A6-
390C5E70E801}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,6
1,00,72,00,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,6
5,00,72,00,66,\
```

00,61,00,63,00,65,00,73,00,5c,00,7b,00,37,00,42,00,32,00,31,0
0,35,00,31,00,\

39,00,39,00,2d,00,41,00,33,00,46,00,33,00,2d,00,34,00,38,00,3
3,00,36,00,2d,\

00,38,00,39,00,41,00,36,00,2d,00,33,00,39,00,30,00,43,00,35,0
0,45,00,37,00,\

30,00,45,00,38,00,30,00,31,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters\Adapters\{A32BB4A3-C9B2-4ADB-A65D-
18BB314BF7F0}]

"LLInterface"=""

"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,6
1,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,6
5,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,41,00,33,00,32,00,42,0
0,42,00,34,00,\

41,00,33,00,2d,00,43,00,39,00,42,00,32,00,2d,00,34,00,41,00,4
4,00,42,00,2d,\

00,41,00,36,00,35,00,44,00,2d,00,31,00,38,00,42,00,42,00,33,0
0,31,00,34,00,\

42,00,46,00,37,00,46,00,30,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters\Adapters\{A71EB7B5-37C6-42DB-BE8F-
BB231FD1BE00}]

"LLInterface"=""

"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,6
1,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,6
5,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,41,00,37,00,31,00,45,0
0,42,00,37,00,\

42,00,35,00,2d,00,33,00,37,00,43,00,36,00,2d,00,34,00,32,00,4
4,00,42,00,2d,\

00,42,00,45,00,38,00,46,00,2d,00,42,00,42,00,32,00,33,00,31,0
0,46,00,44,00,\

31,00,42,00,45,00,30,00,30,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters\Adapters\{BEABCC14-9C0A-4BE9-9817-
14C4092418D3}]

"LLInterface"=""

"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,6
1,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,6
5,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,42,00,45,00,41,00,42,0
0,43,00,43,00,\

31,00,34,00,2d,00,39,00,43,00,30,00,41,00,2d,00,34,00,42,00,4
5,00,39,00,2d,\

00,39,00,38,00,31,00,37,00,2d,00,31,00,34,00,43,00,34,00,30,0
0,39,00,32,00,\

34,00,31,00,38,00,44,00,33,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters\Adapters\{CD3F7746-9E60-4E22-9A40-
7BC6CC6B2E2E}]

"LLInterface"=""

"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,6
1,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,6
5,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,43,00,44,00,33,00,46,0
0,37,00,37,00,\

34,00,36,00,2d,00,39,00,45,00,36,00,30,00,2d,00,34,00,45,00,3
2,00,32,00,2d,\

00,39,00,41,00,34,00,30,00,2d,00,37,00,42,00,43,00,36,00,43,0
0,43,00,36,00,\

42,00,32,00,45,00,32,00,45,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters\DNSRegisteredAdapters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters\Interfaces]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters\Interfaces\{0435C97F-9186-473F-B181-
5449A2CF0042}]

"UseZeroBroadcast"=dword:00000000

"EnableDeadGWDetect"=dword:00000001

"EnableDHCP"=dword:00000000

"IPAddress"=hex(7):31,00,33,00,35,00,2e,00,31,00,2e,00,31,00
,2e,00,31,00,00,00,\

00,00

"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,
00,2e,00,32,00,35,\

00,35,00,2e,00,30,00,00,00,00,00

"DefaultGateway"=hex(7):00,00

"DefaultGatewayMetric"=hex(7):00,00

"NameServer"=""

"Domain"=""

"DisableDynamicUpdate"=dword:00000000

"EnableAdapterDomainNameRegistration"=dword:00000000

"InterfaceMetric"=dword:00000001

"TCPAllowedPorts"=hex(7):30,00,00,00,00,00

"UDPAllowedPorts"=hex(7):30,00,00,00,00,00

"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00

"NTEContextList"=hex(7):00,00

"TcpWindowSize"=dword:00008000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters\Interfaces\{1E07A95A-92A0-4836-BF73-
7AE38F8ACA07}]

"UseZeroBroadcast"=dword:00000000

"EnableDeadGWDetect"=dword:00000001

"EnableDHCP"=dword:00000001

"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00
,00,00,00,00

"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,
00,00,00,00,00

"DefaultGateway"=hex(7):00,00

"DefaultGatewayMetric"=hex(7):00,00

"NameServer"=""

"Domain"=""

"DisableDynamicUpdate"=dword:00000000

"EnableAdapterDomainNameRegistration"=dword:00000000

"InterfaceMetric"=dword:00000001

"TCPAllowedPorts"=hex(7):30,00,00,00,00,00

"UDPAllowedPorts"=hex(7):30,00,00,00,00,00

"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00

"NTEContextList"=hex(7):00,00

"DhcpIPAddress"="0.0.0.0"

"DhcpSubnetMask"="255.0.0.0"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters\Interfaces\{2EA04AA5-93A6-437F-9153-
2F6834D3B795}]

"UseZeroBroadcast"=dword:00000000

"EnableDeadGWDetect"=dword:00000001

"EnableDHCP"=dword:00000000

"IPAddress"=hex(7):31,00,39,00,32,00,2e,00,31,00,36,00,38,00
,2e,00,31,00,31,00,\

2e,00,35,00,31,00,00,00,00,00

"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,
00,2e,00,32,00,35,\

00,35,00,2e,00,30,00,00,00,00,00

"DefaultGateway"=hex(7):00,00

"DefaultGatewayMetric"=hex(7):00,00

"NameServer"=""

"Domain"=""

"DisableDynamicUpdate"=dword:00000000

"EnableAdapterDomainNameRegistration"=dword:00000000

"InterfaceMetric"=dword:00000001

"TCPAllowedPorts"=hex(7):30,00,00,00,00,00

"UDPAllowedPorts"=hex(7):30,00,00,00,00,00

"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00

"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,
30,00,30,00,30,00,\

33,00,00,00,00,00

"DhcpServer"="255.255.255.255"

"Lease"=dword:00000e10

"LeaseObtainedTime"=dword:40b39c06

"T1"=dword:40b3a30e

"T2"=dword:40b3a854

"LeaseTerminatesTime"=dword:40b3aa16

"IPAutoconfigurationAddress"="0.0.0.0"

"IPAutoconfigurationMask"="255.255.0.0"

"IPAutoconfigurationSeed"=dword:00000000

"AddressType"=dword:00000000

"DhcpClassIdBin"=hex:

"TcpWindowSize"=dword:00008000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters\Interfaces\{37430121-7BE3-4B55-8AAB-
D8AD09B2029C}]

"UseZeroBroadcast"=dword:00000000

"EnableDeadGWDetect"=dword:00000001

"EnableDHCP"=dword:00000001

"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00
,00,00,00,00

"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,
00,00,00,00,00

```
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
"NTEContextList"=hex(7):00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{61D33977-710F-4A7E-8D47-4B482BC523F8}]
"UseZeroBroadcast"=dword:00000000
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00
"DefaultGateway"=hex(7):00,00
"EnableDeadGWDetect"=dword:00000001
"DontAddDefaultGateway"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{6FE29D81-59D5-4401-A77E-BE3BC929B6E0}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000001
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
"NTEContextList"=hex(7):00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{7B215199-A3F3-4836-89A6-390C5E70E801}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000001
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
```

```
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
"NTEContextList"=hex(7):00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{9A3AA6AC-5846-407E-8250-F03B604966DC}]
"UseZeroBroadcast"=dword:00000000
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00
"DefaultGateway"=hex(7):00,00
"EnableDeadGWDetect"=dword:00000001
"DontAddDefaultGateway"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A32BB4A3-C9B2-4ADB-A65D-18BB314BF7F0}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,31,00,2e,00,33,00,2e,00,31,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,00,35,00,35,00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
"NTEContextList"=hex(7):00,00
"TcpWindowSize"=dword:00008000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A71EB7B5-37C6-42DB-BE8F-BB231FD1BE00}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,33,00,32,00,2e,00,31,00,2e,00,32,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,00,35,00,35,00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
```

```
"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,\
32,00,00,00,00,00
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:40be0640
"T1"=dword:40be0d48
"T2"=dword:40be128e
"LeaseTerminatesTime"=dword:40be1450
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{BEABCC14-9C0A-4BE9-9817-14C4092418D3}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,31,00,2e,00,32,00,2e,00,31,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,00,35,00,\
00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
"NTEContextList"=hex(7):00,00
"TcpWindowSize"=dword:00008000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{CD3F7746-9E60-4E22-9A40-7BC6CC6B2E2E}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000001
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
"NTEContextList"=hex(7):00,00
"TcpWindowSize"=dword:00008000
"DhcpIPAddress"="0.0.0.0"
"DhcpSubnetMask"="255.0.0.0"
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\PersistentRoutes]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\Tcpip\Parameters\Winsock]
"UseDelayedAcceptance"=dword:00000000
"HelperDllName"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6
d,00,52,00,6f,00,\
6f,00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,3
3,00,32,00,5c,\
00,77,00,73,00,68,00,74,00,63,00,70,00,69,00,70,00,2e,00,64,0
0,6c,00,6c,00,\
00,00
"MaxSockAddrLength"=dword:00000010
"MinSockAddrLength"=dword:00000010
"Mapping"=hex:0b,00,00,00,03,00,00,00,02,00,00,00,01,00,00,0
0,06,00,00,00,02,\
00,00,00,01,00,00,00,00,00,00,02,00,00,00,00,00,00,00,06,0
0,00,00,00,00,\
00,00,00,00,00,06,00,00,00,00,00,00,00,01,00,00,00,06,00,0
0,00,02,00,00,\
00,02,00,00,00,11,00,00,00,02,00,00,00,02,00,00,00,00,00,0
0,02,00,00,00,\
00,00,00,00,11,00,00,00,00,00,00,00,00,00,00,11,00,00,00,0
0,00,00,00,02,\
00,00,00,11,00,00,00,02,00,00,00,03,00,00,00,00,00,00
```

Tpcc software registry.reg

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SOFTWARE\TPCC]
"dlvyLogPath"="c:\inetpub\wwwroot\tpcc\dlvy"
"dlvyQueueLen"=dword:00004e20
"nullDB"=dword:00000000
"htmlTrace"=dword:00000000
"dbName"="tpcc"
"errorLogFile"="c:\inetpub\wwwroot\tpcc\isapi_err.log"
"htmlTraceLogFile"="c:\inetpub\wwwroot\tpcc\isapi.log"
"numUsers"=dword:00005208
"dbType"="DB2"
"dbUserName"="tpcc"
"dbPassword"="tpcc"
"dbInterfacePath"="C:\inetpub\wwwroot\tpcc\db2glue.dll"
"dlvyThreads"=dword:00000005
"isapi_trace"=dword:00000000
```

W3SVC registry.reg

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
```

```
"ImagePath"=hex(2):43,00,3a,00,5c,00,57,00,49,00,4e,00,4e,00
,54,00,5c,00,53,00,\
79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,69,00,6e,00,6
5,00,74,00,73,\
00,72,00,76,00,5c,00,69,00,6e,00,65,00,74,00,69,00,6e,00,66,0
0,6f,00,2e,00,\
65,00,78,00,65,00,00,00
"DisplayName"="World Wide Web Publishing Service"
"DependOnService"=hex(7):49,00,49,00,53,00,41,00,44,00,4d,0
0,49,00,4e,00,00,00,\
00,00
"DependOnGroup"=hex(7):00,00
"ObjectName"="LocalSystem"
"Description"="Provides Web connectivity and administration
through the Internet Information Services snap-in."
"FailureActions"=hex:ff,ff,ff,ff,00,00,00,00,00,00,00,03,00,00,
00,38,c3,0f,\
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
0,00,00,00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\ASP]
"NOTE"="This is for backward compatibility only."
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\ASP\Parameters]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\Parameters]
"MajorVersion"=dword:00000005
"MinorVersion"=dword:00000000
"InstallPath"="C:\WINNT\System32\inetrv"
"CertMapList"="C:\WINNT\System32\inetrv\iiscmap.dll"
"AccessDeniedMessage"="Error: Access is Denied."
"Filter DLLs"=""
"LogFileDirectory"="C:\WINNT\System32\LogFiles"
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\Parameters\ADCLaunch]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\Parameters\ADCLaunch\AdvancedDataFactory]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\Parameters\ADCLaunch\RDSServer.DataFactory]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\Parameters\Script Map]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\Parameters\Virtual Roots]
"/"="c:\inetpub\wwwroot,.201"
"/Scripts"="c:\inetpub\scripts,.204"
"/IISHelp"="c:\winnt\help\iishelp,.201"
"/IISAdmin"="C:\WINNT\System32\inetrv\iisadmin,.201"
"/IISamples"="c:\inetpub\iissamples,.201"
"/MSADC"="c:\program files\common
files\system\msadc,.205"
"/_vti_bin"="C:\Program Files\Common Files\Microsoft
Shared\Web Server Extensions\40\isapi,.205"
"/Printers"="C:\WINNT\web\printers,.201"
"/tpcc"="C:\inetpub\wwwroot\tpcc,.207"
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\Performance]
"Library"="w3ctrs.dll"
"Open"="OpenW3PerformanceData"
"Close"="CloseW3PerformanceData"
```

```
"Collect"="CollectW3PerformanceData"
"Last Counter"=dword:0000008e6
"Last Help"=dword:0000008e7
"First Counter"=dword:000000844
"First Help"=dword:000000845
"Library Validation
Code"=hex:4a,82,e5,79,9f,41,c4,01,10,3d,00,00,00,00,00,00
"WbemAdapFileTime"=hex:00,9a,a9,c0,5f,3f,c4,01
"WbemAdapFileSize"=dword:00003d10
"WbemAdapStatus"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,0
0,30,00,00,00,02,\
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,
00,01,00,00,\
00,00,02,00,70,00,04,00,00,00,00,18,00,fd,01,02,00,01,01,0
0,00,00,00,00,\
05,12,00,00,74,00,6f,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,
00,00,00,05,\
20,00,00,00,20,02,00,00,72,00,73,00,00,00,18,00,8d,01,02,00,0
1,01,00,00,00,\
00,00,05,0b,00,00,00,20,02,00,00,00,1c,00,fd,01,02,00,01,0
2,00,00,00,00,\
00,05,20,00,00,23,02,00,00,72,00,73,00,01,01,00,00,00,00,0
0,05,12,00,00,\
00,01,01,00,00,00,00,05,12,00,00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\Enum]
"0"="Root\LEGACY_W3SVC\00000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

B.3 AIX Parameters

IBM System p5 570

Parameter	Value	Description
Isattr -EI sys0		
SW_dist_intr	false	Enable SW distribution of interrupts
True		
autorestart	true	Automatically REBOOT system after
a crash	True	
boottype	disk	N/A
False		
capacity_inc	1.00	Processor capacity increment
False		
capped	true	Partition is capped
False		
conslogin	enable	System Console Login
False		

```

cpuguard    enable      CPU Guard
True
dedicated   true          Partition is dedicated
False
ent_capacity 16.00        Entitled processor capacity
False
frequency   798000000     System Bus Frequency
False
fullcore    false         Enable full CORE dump
True
fwversion   IBM,SF240_156 Firmware version and revision
levels      False
id_to_partition 0X80000000250600001 Partition ID
False
id_to_system 0X80000000250600000 System ID
False
iostat      false         Continuously maintain DISK I/O history
True
keylock     normal        State of system keylock at boot
time        False
max_capacity 16.00        Maximum potential processor
capacity     False
max_logname 9            Maximum login name length at
boot time   True
maxbuf      20           Maximum number of pages in block
I/O BUFFER CACHE True
maxmbuf     0            Maximum Kbytes of real memory
allowed for MBUFS True
maxpout     0            HIGH water mark for pending write
I/Os per file True
maxuproc    20000       Maximum number of
PROCESSES allowed per user True
min_capacity 1.00        Minimum potential processor
capacity     False
minpout     0            LOW water mark for pending write
I/Os per file True
modelname   IBM,9117-570   Machine name
False
ncargs      6            ARG/ENV list size in 4K byte blocks
True
pre430core  false        Use pre-430 style CORE dump
True
pre520tune  disable      Pre-520 tuning compatibility mode
True
realmem     533987328    Amount of usable physical
memory in Kbytes False
rtasversion 1            Open Firmware RTAS version
False
sed_config  select       Stack Execution Disable (SED)
Mode        True
systemid    IBM,0210007BA Hardware system identifier
False
variable_weight 0        Variable processor capacity weight
False
vmo -L
NAME        CUR DEF BOOT MIN MAX UNIT
TYPE
DEPENDENCIES
-----
cpu_scale_memp 8 8 8 1 64          B

```

```

data_stagger_interval 161 161 161 0 4K-1 4KB
pages D
lgpg_regions
-----
defps 1 1 1 0 1 boolean D
-----
force_realias_lite 0 0 0 0 1 boolean D
-----
framesets 2 2 2 1 10 B
-----
htabscale n/a -1 -1 -4 0 B
-----
kernel_heap_psize 64K 4K 64K 4K 16M bytes
B
lgpg_regions
-----
large_page_heap_size 0 0 0 0 8E-1 bytes
B
lgpg_regions
-----
lgpg_regions 6896 0 6896 0 D
lgpg_size
-----
lgpg_size 16M 0 16M 0 16M bytes
D
lgpg_regions
-----
low_ps_handling 1 1 1 1 2 D
-----
lru_file_repage 1 1 1 0 1 boolean D
-----
lru_poll_interval 10 10 10 0 60000 milliseconds
D
-----
lrubucket 128K 128K 128K 64K 4KB pages
D
-----
maxclient% 90 80 90 1 100 % memory
D
maxperm%
minperm%
-----
maxfree 1088 1088 1088 8 200K 4KB pages
D
minfree
memory_frames
-----
maxperm 3151K 3151K S
-----
maxperm% 90 80 90 1 100 % memory
D
minperm%
maxclient%
-----
maxpin 130332K 130332K
S
-----
maxpin% 99 80 99 1 99 % memory
D
pinnable_frames
memory_frames
-----

```

```

mbuf_heap_psize 64K 0 0 0 16M bytes
B
-----
memory_affinity 1 1 1 0 1 boolean
B
-----
memory_frames 127M 127M 4KB pages
S
-----
memplace_data 2 2 2 1 2 D
memory_affinity
-----
memplace_mapped_file 2 2 2 1 2
D
memory_affinity
-----
memplace_shm_anonymous 2 2 2 1 2
D
memory_affinity
-----
memplace_shm_named 2 2 2 1 2
D
memory_affinity
-----
memplace_stack 2 2 2 1 2 D
memory_affinity
-----
memplace_text 2 2 2 1 2 D
memory_affinity
-----
memplace_unmapped_file 2 2 2 1 2
D
memory_affinity
-----
mempools 0 0 d
cpu_scale_memp
-----
minfree 960 960 960 8 200K 4KB pages
D
maxfree
memory_frames
-----
minperm 717079 717079 S
-----
minperm% 20 20 20 1 100 % memory
D
maxperm%
maxclient%
-----
nokilluid 0 0 0 0 4G-1 uid D
-----
npskill 16896 16896 16896 1 2M-1 4KB pages
D
-----
npsrpgmax 132K 132K 132K 0 2M-1 4KB
pages D
npsrpgmin
-----
npsrpgmin 99K 99K 99K 0 2M-1 4KB pages
D
npsrpgmax
-----

```



```

npsscubmax      132K 132K 132K 0  2M-1 4KB
pages          D
npsscubmin
-----
npsscubmin      99K  99K  99K  0  2M-1 4KB
pages          D
npsscubmax
-----
npswarn         66K  66K  66K  0  2M-1 4KB pages
D
-----
num_spec_dataseg  0  0  0  0  B
-----
numpsblks       2112K  2112K  4KB blocks
S
-----
page_steal_method  0  0  0  0  1  boolean
B
-----
pagecoloring     n/a  0  0  0  1  boolean  B
-----
pinnable_frames  2777K  2777K  4KB pages
S
-----
pta_balance_threshold n/a  1  1  0  99  % pta
segment        D
-----
relalias_percentage  0  0  0  0  32K-1
D
-----
rpgclean         0  0  0  0  1  boolean  D
-----
rpgcontrol       2  2  2  0  3  D
-----
scrub             0  0  0  0  1  boolean  D
-----
scrubclean       0  0  0  0  1  boolean  D
-----
soft_min_lgpgs_vmpool  0  0  0  0  90  %
D
lgpg_regions
-----
spec_dataseg_int  512  512  512  0  B
-----
strict_maxclient  1  1  1  0  1  boolean  D
strict_maxperm
-----
strict_maxperm   0  0  0  0  1  boolean
D
strict_maxclient
-----
v_pinshm         1  0  1  0  1  boolean  D
-----
vm_modlist_threshold -1 -1 -1 -2 2G-1
D
-----
vmm_fork_policy  1  1  1  0  1  boolean
D
-----
vmm_mpsize_support  1  1  1  0  1  boolean
B

```

```

-----
ioo -L
NAME           CUR  DEF  BOOT  MIN  MAX  UNIT
TYPE
DEPENDENCIES
-----
j2_dynamicBufferPreallocation
          16  16  16  0  256  16K slabs  D
-----
j2_inodeCacheSize  400  400  400  1  1000
D
-----
j2_maxPageReadAhead  128  128  128  0  64K  4KB
pages  D
-----
j2_maxRandomWrite  0  0  0  0  64K  4KB
pages  D
-----
j2_maxUsableMaxTransfer  512  512  512  1  4K
pages  M
-----
j2_metadataCacheSize  400  400  400  1  1000
D
-----
j2_minPageReadAhead  2  2  2  0  64K  4KB
pages  D
-----
j2_nBufferPerPagerDevice  512  512  512  0  256K
M
-----
j2_nPagesPerWriteBehindCluster
          32  32  32  0  64K  D
-----
j2_nRandomCluster  0  0  0  0  64K  16KB
clusters  D
-----
j2_nonFatalCrashesSystem  0  0  0  0  1  boolean
D
-----
j2_syncModifiedMapped  1  1  1  0  1  boolean
D
-----
jfs_cread_enabled  0  0  0  0  1  boolean
D
-----
jfs_use_read_lock  1  1  1  0  1  boolean
D
-----
lvm_bufcnt       9  9  9  1  64  128KB/buffer
D
-----
maxpgahead       8  8  8  0  4K  4KB pages
D
minpgahead
-----
maxrandwrt       0  0  0  0  512K  4KB pages
D
-----
memory_frames    127M  127M  4KB pages
S
-----

```

```

minpgahead       2  2  2  0  4K  4KB pages
D
maxpgahead
-----
numclust         1  1  1  0  2G-1  16KB/cluster
D
-----
numfsbufs        196  196  196  1  2G-1
M
-----
pd_npages        64K  64K  64K  1  512K  4KB
pages  D
-----
pgahd_scale_thresh  0  0  0  0  104294K  4KB
pages  D
-----
pv_min_pbuf      512  512  512  512  2G-1
D
-----
sync_release_ilock  0  0  0  0  1  boolean
D
-----

```



```

alter tablespace ts_ware_024 prefetchsize 4096;
alter tablespace ts_ware_025 prefetchsize 4096;
alter tablespace ts_ware_026 prefetchsize 4096;
alter tablespace ts_ware_027 prefetchsize 4096;
alter tablespace ts_ware_028 prefetchsize 4096;
alter tablespace ts_ware_029 prefetchsize 4096;
alter tablespace ts_ware_030 prefetchsize 4096;
alter tablespace ts_ware_031 prefetchsize 4096;
alter tablespace ts_ware_032 prefetchsize 4096;
alter tablespace ts_ware_033 prefetchsize 4096;
alter tablespace ts_ware_034 prefetchsize 4096;
alter tablespace ts_ware_035 prefetchsize 4096;
alter tablespace ts_ware_036 prefetchsize 4096;
alter tablespace ts_ware_037 prefetchsize 4096;
alter tablespace ts_ware_038 prefetchsize 4096;
alter tablespace ts_ware_039 prefetchsize 4096;
alter tablespace ts_ware_040 prefetchsize 4096;
alter tablespace ts_ware_041 prefetchsize 4096;
alter tablespace ts_ware_042 prefetchsize 4096;
alter tablespace ts_ware_043 prefetchsize 4096;
alter tablespace ts_ware_044 prefetchsize 4096;
alter tablespace ts_ware_045 prefetchsize 4096;
alter tablespace ts_ware_046 prefetchsize 4096;
alter tablespace ts_ware_047 prefetchsize 4096;
alter tablespace ts_ware_048 prefetchsize 4096;
alter tablespace ts_ware_049 prefetchsize 4096;
alter tablespace ts_ware_050 prefetchsize 4096;
alter tablespace ts_ware_051 prefetchsize 4096;
connect reset;

```

DDL/CRCONST CUSTOMER.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER1 OFF;
ALTER TABLE CUSTOMER1 DROP CONSTRAINT
CUSTOMER1CKC;
ALTER TABLE CUSTOMER1 ADD CONSTRAINT
CUSTOMER1CKC CHECK (C_W_ID BETWEEN 1 AND 1600);
SET INTEGRITY FOR CUSTOMER1 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER2 OFF;
ALTER TABLE CUSTOMER2 DROP CONSTRAINT
CUSTOMER2CKC;
ALTER TABLE CUSTOMER2 ADD CONSTRAINT
CUSTOMER2CKC CHECK (C_W_ID BETWEEN 1601 AND
3200);
SET INTEGRITY FOR CUSTOMER2 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER3 OFF;
ALTER TABLE CUSTOMER3 DROP CONSTRAINT
CUSTOMER3CKC;
ALTER TABLE CUSTOMER3 ADD CONSTRAINT
CUSTOMER3CKC CHECK (C_W_ID BETWEEN 3201 AND
4800);
SET INTEGRITY FOR CUSTOMER3 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

```

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER4 OFF;
ALTER TABLE CUSTOMER4 DROP CONSTRAINT
CUSTOMER4CKC;
ALTER TABLE CUSTOMER4 ADD CONSTRAINT
CUSTOMER4CKC CHECK (C_W_ID BETWEEN 4801 AND
6400);
SET INTEGRITY FOR CUSTOMER4 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER5 OFF;
ALTER TABLE CUSTOMER5 DROP CONSTRAINT
CUSTOMER5CKC;
ALTER TABLE CUSTOMER5 ADD CONSTRAINT
CUSTOMER5CKC CHECK (C_W_ID BETWEEN 6401 AND
8000);
SET INTEGRITY FOR CUSTOMER5 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER6 OFF;
ALTER TABLE CUSTOMER6 DROP CONSTRAINT
CUSTOMER6CKC;
ALTER TABLE CUSTOMER6 ADD CONSTRAINT
CUSTOMER6CKC CHECK (C_W_ID BETWEEN 8001 AND
9600);
SET INTEGRITY FOR CUSTOMER6 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER7 OFF;
ALTER TABLE CUSTOMER7 DROP CONSTRAINT
CUSTOMER7CKC;
ALTER TABLE CUSTOMER7 ADD CONSTRAINT
CUSTOMER7CKC CHECK (C_W_ID BETWEEN 9601 AND
11200);
SET INTEGRITY FOR CUSTOMER7 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER8 OFF;
ALTER TABLE CUSTOMER8 DROP CONSTRAINT
CUSTOMER8CKC;
ALTER TABLE CUSTOMER8 ADD CONSTRAINT
CUSTOMER8CKC CHECK (C_W_ID BETWEEN 11201 AND
12800);
SET INTEGRITY FOR CUSTOMER8 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER9 OFF;
ALTER TABLE CUSTOMER9 DROP CONSTRAINT
CUSTOMER9CKC;
ALTER TABLE CUSTOMER9 ADD CONSTRAINT
CUSTOMER9CKC CHECK (C_W_ID BETWEEN 12801 AND
14400);
SET INTEGRITY FOR CUSTOMER9 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER10 OFF;

```

```

ALTER TABLE CUSTOMER10 DROP CONSTRAINT
CUSTOMER10CKC;
ALTER TABLE CUSTOMER10 ADD CONSTRAINT
CUSTOMER10CKC CHECK (C_W_ID BETWEEN 14401 AND
16000);
SET INTEGRITY FOR CUSTOMER10 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER11 OFF;
ALTER TABLE CUSTOMER11 DROP CONSTRAINT
CUSTOMER11CKC;
ALTER TABLE CUSTOMER11 ADD CONSTRAINT
CUSTOMER11CKC CHECK (C_W_ID BETWEEN 16001 AND
17600);
SET INTEGRITY FOR CUSTOMER11 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER12 OFF;
ALTER TABLE CUSTOMER12 DROP CONSTRAINT
CUSTOMER12CKC;
ALTER TABLE CUSTOMER12 ADD CONSTRAINT
CUSTOMER12CKC CHECK (C_W_ID BETWEEN 17601 AND
19200);
SET INTEGRITY FOR CUSTOMER12 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER13 OFF;
ALTER TABLE CUSTOMER13 DROP CONSTRAINT
CUSTOMER13CKC;
ALTER TABLE CUSTOMER13 ADD CONSTRAINT
CUSTOMER13CKC CHECK (C_W_ID BETWEEN 19201 AND
20800);
SET INTEGRITY FOR CUSTOMER13 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER14 OFF;
ALTER TABLE CUSTOMER14 DROP CONSTRAINT
CUSTOMER14CKC;
ALTER TABLE CUSTOMER14 ADD CONSTRAINT
CUSTOMER14CKC CHECK (C_W_ID BETWEEN 20801 AND
22400);
SET INTEGRITY FOR CUSTOMER14 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER15 OFF;
ALTER TABLE CUSTOMER15 DROP CONSTRAINT
CUSTOMER15CKC;
ALTER TABLE CUSTOMER15 ADD CONSTRAINT
CUSTOMER15CKC CHECK (C_W_ID BETWEEN 22401 AND
24000);
SET INTEGRITY FOR CUSTOMER15 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER16 OFF;
ALTER TABLE CUSTOMER16 DROP CONSTRAINT
CUSTOMER16CKC;

```



```

ALTER TABLE CUSTOMER16 ADD CONSTRAINT
CUSTOMER16CKC CHECK (C_W_ID BETWEEN 24001 AND
25600);
SET INTEGRITY FOR CUSTOMER16 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER17 OFF;
ALTER TABLE CUSTOMER17 DROP CONSTRAINT
CUSTOMER17CKC;
ALTER TABLE CUSTOMER17 ADD CONSTRAINT
CUSTOMER17CKC CHECK (C_W_ID BETWEEN 25601 AND
27200);
SET INTEGRITY FOR CUSTOMER17 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER18 OFF;
ALTER TABLE CUSTOMER18 DROP CONSTRAINT
CUSTOMER18CKC;
ALTER TABLE CUSTOMER18 ADD CONSTRAINT
CUSTOMER18CKC CHECK (C_W_ID BETWEEN 27201 AND
28800);
SET INTEGRITY FOR CUSTOMER18 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER19 OFF;
ALTER TABLE CUSTOMER19 DROP CONSTRAINT
CUSTOMER19CKC;
ALTER TABLE CUSTOMER19 ADD CONSTRAINT
CUSTOMER19CKC CHECK (C_W_ID BETWEEN 28801 AND
30400);
SET INTEGRITY FOR CUSTOMER19 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER20 OFF;
ALTER TABLE CUSTOMER20 DROP CONSTRAINT
CUSTOMER20CKC;
ALTER TABLE CUSTOMER20 ADD CONSTRAINT
CUSTOMER20CKC CHECK (C_W_ID BETWEEN 30401 AND
32000);
SET INTEGRITY FOR CUSTOMER20 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER21 OFF;
ALTER TABLE CUSTOMER21 DROP CONSTRAINT
CUSTOMER21CKC;
ALTER TABLE CUSTOMER21 ADD CONSTRAINT
CUSTOMER21CKC CHECK (C_W_ID BETWEEN 32001 AND
33600);
SET INTEGRITY FOR CUSTOMER21 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER22 OFF;
ALTER TABLE CUSTOMER22 DROP CONSTRAINT
CUSTOMER22CKC;

```

```

ALTER TABLE CUSTOMER22 ADD CONSTRAINT
CUSTOMER22CKC CHECK (C_W_ID BETWEEN 33601 AND
35200);
SET INTEGRITY FOR CUSTOMER22 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER23 OFF;
ALTER TABLE CUSTOMER23 DROP CONSTRAINT
CUSTOMER23CKC;
ALTER TABLE CUSTOMER23 ADD CONSTRAINT
CUSTOMER23CKC CHECK (C_W_ID BETWEEN 35201 AND
36800);
SET INTEGRITY FOR CUSTOMER23 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER24 OFF;
ALTER TABLE CUSTOMER24 DROP CONSTRAINT
CUSTOMER24CKC;
ALTER TABLE CUSTOMER24 ADD CONSTRAINT
CUSTOMER24CKC CHECK (C_W_ID BETWEEN 36801 AND
38400);
SET INTEGRITY FOR CUSTOMER24 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER25 OFF;
ALTER TABLE CUSTOMER25 DROP CONSTRAINT
CUSTOMER25CKC;
ALTER TABLE CUSTOMER25 ADD CONSTRAINT
CUSTOMER25CKC CHECK (C_W_ID BETWEEN 38401 AND
40000);
SET INTEGRITY FOR CUSTOMER25 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER26 OFF;
ALTER TABLE CUSTOMER26 DROP CONSTRAINT
CUSTOMER26CKC;
ALTER TABLE CUSTOMER26 ADD CONSTRAINT
CUSTOMER26CKC CHECK (C_W_ID BETWEEN 40001 AND
41600);
SET INTEGRITY FOR CUSTOMER26 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER27 OFF;
ALTER TABLE CUSTOMER27 DROP CONSTRAINT
CUSTOMER27CKC;
ALTER TABLE CUSTOMER27 ADD CONSTRAINT
CUSTOMER27CKC CHECK (C_W_ID BETWEEN 41601 AND
43200);
SET INTEGRITY FOR CUSTOMER27 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER28 OFF;
ALTER TABLE CUSTOMER28 DROP CONSTRAINT
CUSTOMER28CKC;

```

```

ALTER TABLE CUSTOMER28 ADD CONSTRAINT
CUSTOMER28CKC CHECK (C_W_ID BETWEEN 43201 AND
44800);
SET INTEGRITY FOR CUSTOMER28 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER29 OFF;
ALTER TABLE CUSTOMER29 DROP CONSTRAINT
CUSTOMER29CKC;
ALTER TABLE CUSTOMER29 ADD CONSTRAINT
CUSTOMER29CKC CHECK (C_W_ID BETWEEN 44801 AND
46400);
SET INTEGRITY FOR CUSTOMER29 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER30 OFF;
ALTER TABLE CUSTOMER30 DROP CONSTRAINT
CUSTOMER30CKC;
ALTER TABLE CUSTOMER30 ADD CONSTRAINT
CUSTOMER30CKC CHECK (C_W_ID BETWEEN 46401 AND
48000);
SET INTEGRITY FOR CUSTOMER30 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER31 OFF;
ALTER TABLE CUSTOMER31 DROP CONSTRAINT
CUSTOMER31CKC;
ALTER TABLE CUSTOMER31 ADD CONSTRAINT
CUSTOMER31CKC CHECK (C_W_ID BETWEEN 48001 AND
49600);
SET INTEGRITY FOR CUSTOMER31 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER32 OFF;
ALTER TABLE CUSTOMER32 DROP CONSTRAINT
CUSTOMER32CKC;
ALTER TABLE CUSTOMER32 ADD CONSTRAINT
CUSTOMER32CKC CHECK (C_W_ID BETWEEN 49601 AND
51200);
SET INTEGRITY FOR CUSTOMER32 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER33 OFF;
ALTER TABLE CUSTOMER33 DROP CONSTRAINT
CUSTOMER33CKC;
ALTER TABLE CUSTOMER33 ADD CONSTRAINT
CUSTOMER33CKC CHECK (C_W_ID BETWEEN 51201 AND
52800);
SET INTEGRITY FOR CUSTOMER33 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER34 OFF;
ALTER TABLE CUSTOMER34 DROP CONSTRAINT
CUSTOMER34CKC;

```

```

ALTER TABLE CUSTOMER34 ADD CONSTRAINT
CUSTOMER34CKC CHECK (C_W_ID BETWEEN 52801 AND
54400);
SET INTEGRITY FOR CUSTOMER34 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER35 OFF;
ALTER TABLE CUSTOMER35 DROP CONSTRAINT
CUSTOMER35CKC;
ALTER TABLE CUSTOMER35 ADD CONSTRAINT
CUSTOMER35CKC CHECK (C_W_ID BETWEEN 54401 AND
56000);
SET INTEGRITY FOR CUSTOMER35 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER36 OFF;
ALTER TABLE CUSTOMER36 DROP CONSTRAINT
CUSTOMER36CKC;
ALTER TABLE CUSTOMER36 ADD CONSTRAINT
CUSTOMER36CKC CHECK (C_W_ID BETWEEN 56001 AND
57600);
SET INTEGRITY FOR CUSTOMER36 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER37 OFF;
ALTER TABLE CUSTOMER37 DROP CONSTRAINT
CUSTOMER37CKC;
ALTER TABLE CUSTOMER37 ADD CONSTRAINT
CUSTOMER37CKC CHECK (C_W_ID BETWEEN 57601 AND
59200);
SET INTEGRITY FOR CUSTOMER37 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER38 OFF;
ALTER TABLE CUSTOMER38 DROP CONSTRAINT
CUSTOMER38CKC;
ALTER TABLE CUSTOMER38 ADD CONSTRAINT
CUSTOMER38CKC CHECK (C_W_ID BETWEEN 59201 AND
60800);
SET INTEGRITY FOR CUSTOMER38 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER39 OFF;
ALTER TABLE CUSTOMER39 DROP CONSTRAINT
CUSTOMER39CKC;
ALTER TABLE CUSTOMER39 ADD CONSTRAINT
CUSTOMER39CKC CHECK (C_W_ID BETWEEN 60801 AND
62400);
SET INTEGRITY FOR CUSTOMER39 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER40 OFF;
ALTER TABLE CUSTOMER40 DROP CONSTRAINT
CUSTOMER40CKC;

```

```

ALTER TABLE CUSTOMER40 ADD CONSTRAINT
CUSTOMER40CKC CHECK (C_W_ID BETWEEN 62401 AND
64000);
SET INTEGRITY FOR CUSTOMER40 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER41 OFF;
ALTER TABLE CUSTOMER41 DROP CONSTRAINT
CUSTOMER41CKC;
ALTER TABLE CUSTOMER41 ADD CONSTRAINT
CUSTOMER41CKC CHECK (C_W_ID BETWEEN 64001 AND
65600);
SET INTEGRITY FOR CUSTOMER41 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER42 OFF;
ALTER TABLE CUSTOMER42 DROP CONSTRAINT
CUSTOMER42CKC;
ALTER TABLE CUSTOMER42 ADD CONSTRAINT
CUSTOMER42CKC CHECK (C_W_ID BETWEEN 65601 AND
67200);
SET INTEGRITY FOR CUSTOMER42 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER43 OFF;
ALTER TABLE CUSTOMER43 DROP CONSTRAINT
CUSTOMER43CKC;
ALTER TABLE CUSTOMER43 ADD CONSTRAINT
CUSTOMER43CKC CHECK (C_W_ID BETWEEN 67201 AND
68800);
SET INTEGRITY FOR CUSTOMER43 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER44 OFF;
ALTER TABLE CUSTOMER44 DROP CONSTRAINT
CUSTOMER44CKC;
ALTER TABLE CUSTOMER44 ADD CONSTRAINT
CUSTOMER44CKC CHECK (C_W_ID BETWEEN 68801 AND
70400);
SET INTEGRITY FOR CUSTOMER44 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER45 OFF;
ALTER TABLE CUSTOMER45 DROP CONSTRAINT
CUSTOMER45CKC;
ALTER TABLE CUSTOMER45 ADD CONSTRAINT
CUSTOMER45CKC CHECK (C_W_ID BETWEEN 70401 AND
72000);
SET INTEGRITY FOR CUSTOMER45 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER46 OFF;
ALTER TABLE CUSTOMER46 DROP CONSTRAINT
CUSTOMER46CKC;

```

```

ALTER TABLE CUSTOMER46 ADD CONSTRAINT
CUSTOMER46CKC CHECK (C_W_ID BETWEEN 72001 AND
73600);
SET INTEGRITY FOR CUSTOMER46 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER47 OFF;
ALTER TABLE CUSTOMER47 DROP CONSTRAINT
CUSTOMER47CKC;
ALTER TABLE CUSTOMER47 ADD CONSTRAINT
CUSTOMER47CKC CHECK (C_W_ID BETWEEN 73601 AND
75200);
SET INTEGRITY FOR CUSTOMER47 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER48 OFF;
ALTER TABLE CUSTOMER48 DROP CONSTRAINT
CUSTOMER48CKC;
ALTER TABLE CUSTOMER48 ADD CONSTRAINT
CUSTOMER48CKC CHECK (C_W_ID BETWEEN 75201 AND
76800);
SET INTEGRITY FOR CUSTOMER48 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER49 OFF;
ALTER TABLE CUSTOMER49 DROP CONSTRAINT
CUSTOMER49CKC;
ALTER TABLE CUSTOMER49 ADD CONSTRAINT
CUSTOMER49CKC CHECK (C_W_ID BETWEEN 76801 AND
78400);
SET INTEGRITY FOR CUSTOMER49 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER50 OFF;
ALTER TABLE CUSTOMER50 DROP CONSTRAINT
CUSTOMER50CKC;
ALTER TABLE CUSTOMER50 ADD CONSTRAINT
CUSTOMER50CKC CHECK (C_W_ID BETWEEN 78401 AND
80000);
SET INTEGRITY FOR CUSTOMER50 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER51 OFF;
ALTER TABLE CUSTOMER51 DROP CONSTRAINT
CUSTOMER51CKC;
ALTER TABLE CUSTOMER51 ADD CONSTRAINT
CUSTOMER51CKC CHECK (C_W_ID >= 80001);
SET INTEGRITY FOR CUSTOMER51 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

DDL/CRCONST_DISTRICT.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT1 OFF;

```

```

ALTER TABLE DISTRICT1 DROP CONSTRAINT
DISTRICT1CKC;
ALTER TABLE DISTRICT1 ADD CONSTRAINT
DISTRICT1CKC CHECK (D_W_ID BETWEEN 1 AND 1600);
SET INTEGRITY FOR DISTRICT1 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT2 OFF;
ALTER TABLE DISTRICT2 DROP CONSTRAINT
DISTRICT2CKC;
ALTER TABLE DISTRICT2 ADD CONSTRAINT
DISTRICT2CKC CHECK (D_W_ID BETWEEN 1601 AND
3200);
SET INTEGRITY FOR DISTRICT2 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT3 OFF;
ALTER TABLE DISTRICT3 DROP CONSTRAINT
DISTRICT3CKC;
ALTER TABLE DISTRICT3 ADD CONSTRAINT
DISTRICT3CKC CHECK (D_W_ID BETWEEN 3201 AND
4800);
SET INTEGRITY FOR DISTRICT3 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT4 OFF;
ALTER TABLE DISTRICT4 DROP CONSTRAINT
DISTRICT4CKC;
ALTER TABLE DISTRICT4 ADD CONSTRAINT
DISTRICT4CKC CHECK (D_W_ID BETWEEN 4801 AND
6400);
SET INTEGRITY FOR DISTRICT4 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT5 OFF;
ALTER TABLE DISTRICT5 DROP CONSTRAINT
DISTRICT5CKC;
ALTER TABLE DISTRICT5 ADD CONSTRAINT
DISTRICT5CKC CHECK (D_W_ID BETWEEN 6401 AND
8000);
SET INTEGRITY FOR DISTRICT5 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT6 OFF;
ALTER TABLE DISTRICT6 DROP CONSTRAINT
DISTRICT6CKC;
ALTER TABLE DISTRICT6 ADD CONSTRAINT
DISTRICT6CKC CHECK (D_W_ID BETWEEN 8001 AND
9600);
SET INTEGRITY FOR DISTRICT6 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT7 OFF;
ALTER TABLE DISTRICT7 DROP CONSTRAINT
DISTRICT7CKC;

```

```

ALTER TABLE DISTRICT7 ADD CONSTRAINT
DISTRICT7CKC CHECK (D_W_ID BETWEEN 9601 AND
11200);
SET INTEGRITY FOR DISTRICT7 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT8 OFF;
ALTER TABLE DISTRICT8 DROP CONSTRAINT
DISTRICT8CKC;
ALTER TABLE DISTRICT8 ADD CONSTRAINT
DISTRICT8CKC CHECK (D_W_ID BETWEEN 11201 AND
12800);
SET INTEGRITY FOR DISTRICT8 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT9 OFF;
ALTER TABLE DISTRICT9 DROP CONSTRAINT
DISTRICT9CKC;
ALTER TABLE DISTRICT9 ADD CONSTRAINT
DISTRICT9CKC CHECK (D_W_ID BETWEEN 12801 AND
14400);
SET INTEGRITY FOR DISTRICT9 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT10 OFF;
ALTER TABLE DISTRICT10 DROP CONSTRAINT
DISTRICT10CKC;
ALTER TABLE DISTRICT10 ADD CONSTRAINT
DISTRICT10CKC CHECK (D_W_ID BETWEEN 14401 AND
16000);
SET INTEGRITY FOR DISTRICT10 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT11 OFF;
ALTER TABLE DISTRICT11 DROP CONSTRAINT
DISTRICT11CKC;
ALTER TABLE DISTRICT11 ADD CONSTRAINT
DISTRICT11CKC CHECK (D_W_ID BETWEEN 16001 AND
17600);
SET INTEGRITY FOR DISTRICT11 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT12 OFF;
ALTER TABLE DISTRICT12 DROP CONSTRAINT
DISTRICT12CKC;
ALTER TABLE DISTRICT12 ADD CONSTRAINT
DISTRICT12CKC CHECK (D_W_ID BETWEEN 17601 AND
19200);
SET INTEGRITY FOR DISTRICT12 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT13 OFF;
ALTER TABLE DISTRICT13 DROP CONSTRAINT
DISTRICT13CKC;

```

```

ALTER TABLE DISTRICT13 ADD CONSTRAINT
DISTRICT13CKC CHECK (D_W_ID BETWEEN 19201 AND
20800);
SET INTEGRITY FOR DISTRICT13 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT14 OFF;
ALTER TABLE DISTRICT14 DROP CONSTRAINT
DISTRICT14CKC;
ALTER TABLE DISTRICT14 ADD CONSTRAINT
DISTRICT14CKC CHECK (D_W_ID BETWEEN 20801 AND
22400);
SET INTEGRITY FOR DISTRICT14 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT15 OFF;
ALTER TABLE DISTRICT15 DROP CONSTRAINT
DISTRICT15CKC;
ALTER TABLE DISTRICT15 ADD CONSTRAINT
DISTRICT15CKC CHECK (D_W_ID BETWEEN 22401 AND
24000);
SET INTEGRITY FOR DISTRICT15 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT16 OFF;
ALTER TABLE DISTRICT16 DROP CONSTRAINT
DISTRICT16CKC;
ALTER TABLE DISTRICT16 ADD CONSTRAINT
DISTRICT16CKC CHECK (D_W_ID BETWEEN 24001 AND
25600);
SET INTEGRITY FOR DISTRICT16 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT17 OFF;
ALTER TABLE DISTRICT17 DROP CONSTRAINT
DISTRICT17CKC;
ALTER TABLE DISTRICT17 ADD CONSTRAINT
DISTRICT17CKC CHECK (D_W_ID BETWEEN 25601 AND
27200);
SET INTEGRITY FOR DISTRICT17 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT18 OFF;
ALTER TABLE DISTRICT18 DROP CONSTRAINT
DISTRICT18CKC;
ALTER TABLE DISTRICT18 ADD CONSTRAINT
DISTRICT18CKC CHECK (D_W_ID BETWEEN 27201 AND
28800);
SET INTEGRITY FOR DISTRICT18 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT19 OFF;
ALTER TABLE DISTRICT19 DROP CONSTRAINT
DISTRICT19CKC;

```

```

ALTER TABLE DISTRICT19 ADD CONSTRAINT
DISTRICT19CKC CHECK (D_W_ID BETWEEN 28801 AND
30400);
SET INTEGRITY FOR DISTRICT19 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT20 OFF;
ALTER TABLE DISTRICT20 DROP CONSTRAINT
DISTRICT20CKC;
ALTER TABLE DISTRICT20 ADD CONSTRAINT
DISTRICT20CKC CHECK (D_W_ID BETWEEN 30401 AND
32000);
SET INTEGRITY FOR DISTRICT20 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT21 OFF;
ALTER TABLE DISTRICT21 DROP CONSTRAINT
DISTRICT21CKC;
ALTER TABLE DISTRICT21 ADD CONSTRAINT
DISTRICT21CKC CHECK (D_W_ID BETWEEN 32001 AND
33600);
SET INTEGRITY FOR DISTRICT21 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT22 OFF;
ALTER TABLE DISTRICT22 DROP CONSTRAINT
DISTRICT22CKC;
ALTER TABLE DISTRICT22 ADD CONSTRAINT
DISTRICT22CKC CHECK (D_W_ID BETWEEN 33601 AND
35200);
SET INTEGRITY FOR DISTRICT22 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT23 OFF;
ALTER TABLE DISTRICT23 DROP CONSTRAINT
DISTRICT23CKC;
ALTER TABLE DISTRICT23 ADD CONSTRAINT
DISTRICT23CKC CHECK (D_W_ID BETWEEN 35201 AND
36800);
SET INTEGRITY FOR DISTRICT23 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT24 OFF;
ALTER TABLE DISTRICT24 DROP CONSTRAINT
DISTRICT24CKC;
ALTER TABLE DISTRICT24 ADD CONSTRAINT
DISTRICT24CKC CHECK (D_W_ID BETWEEN 36801 AND
38400);
SET INTEGRITY FOR DISTRICT24 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT25 OFF;
ALTER TABLE DISTRICT25 DROP CONSTRAINT
DISTRICT25CKC;

```

```

ALTER TABLE DISTRICT25 ADD CONSTRAINT
DISTRICT25CKC CHECK (D_W_ID BETWEEN 38401 AND
40000);
SET INTEGRITY FOR DISTRICT25 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT26 OFF;
ALTER TABLE DISTRICT26 DROP CONSTRAINT
DISTRICT26CKC;
ALTER TABLE DISTRICT26 ADD CONSTRAINT
DISTRICT26CKC CHECK (D_W_ID BETWEEN 40001 AND
41600);
SET INTEGRITY FOR DISTRICT26 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT27 OFF;
ALTER TABLE DISTRICT27 DROP CONSTRAINT
DISTRICT27CKC;
ALTER TABLE DISTRICT27 ADD CONSTRAINT
DISTRICT27CKC CHECK (D_W_ID BETWEEN 41601 AND
43200);
SET INTEGRITY FOR DISTRICT27 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT28 OFF;
ALTER TABLE DISTRICT28 DROP CONSTRAINT
DISTRICT28CKC;
ALTER TABLE DISTRICT28 ADD CONSTRAINT
DISTRICT28CKC CHECK (D_W_ID BETWEEN 43201 AND
44800);
SET INTEGRITY FOR DISTRICT28 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT29 OFF;
ALTER TABLE DISTRICT29 DROP CONSTRAINT
DISTRICT29CKC;
ALTER TABLE DISTRICT29 ADD CONSTRAINT
DISTRICT29CKC CHECK (D_W_ID BETWEEN 44801 AND
46400);
SET INTEGRITY FOR DISTRICT29 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT30 OFF;
ALTER TABLE DISTRICT30 DROP CONSTRAINT
DISTRICT30CKC;
ALTER TABLE DISTRICT30 ADD CONSTRAINT
DISTRICT30CKC CHECK (D_W_ID BETWEEN 46401 AND
48000);
SET INTEGRITY FOR DISTRICT30 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT31 OFF;
ALTER TABLE DISTRICT31 DROP CONSTRAINT
DISTRICT31CKC;

```

```

ALTER TABLE DISTRICT31 ADD CONSTRAINT
DISTRICT31CKC CHECK (D_W_ID BETWEEN 48001 AND
49600);
SET INTEGRITY FOR DISTRICT31 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT32 OFF;
ALTER TABLE DISTRICT32 DROP CONSTRAINT
DISTRICT32CKC;
ALTER TABLE DISTRICT32 ADD CONSTRAINT
DISTRICT32CKC CHECK (D_W_ID BETWEEN 49601 AND
51200);
SET INTEGRITY FOR DISTRICT32 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT33 OFF;
ALTER TABLE DISTRICT33 DROP CONSTRAINT
DISTRICT33CKC;
ALTER TABLE DISTRICT33 ADD CONSTRAINT
DISTRICT33CKC CHECK (D_W_ID BETWEEN 51201 AND
52800);
SET INTEGRITY FOR DISTRICT33 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT34 OFF;
ALTER TABLE DISTRICT34 DROP CONSTRAINT
DISTRICT34CKC;
ALTER TABLE DISTRICT34 ADD CONSTRAINT
DISTRICT34CKC CHECK (D_W_ID BETWEEN 52801 AND
54400);
SET INTEGRITY FOR DISTRICT34 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT35 OFF;
ALTER TABLE DISTRICT35 DROP CONSTRAINT
DISTRICT35CKC;
ALTER TABLE DISTRICT35 ADD CONSTRAINT
DISTRICT35CKC CHECK (D_W_ID BETWEEN 54401 AND
56000);
SET INTEGRITY FOR DISTRICT35 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT36 OFF;
ALTER TABLE DISTRICT36 DROP CONSTRAINT
DISTRICT36CKC;
ALTER TABLE DISTRICT36 ADD CONSTRAINT
DISTRICT36CKC CHECK (D_W_ID BETWEEN 56001 AND
57600);
SET INTEGRITY FOR DISTRICT36 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT37 OFF;
ALTER TABLE DISTRICT37 DROP CONSTRAINT
DISTRICT37CKC;

```

```

ALTER TABLE DISTRICT37 ADD CONSTRAINT
DISTRICT37CKC CHECK (D_W_ID BETWEEN 57601 AND
59200);
SET INTEGRITY FOR DISTRICT37 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT38 OFF;
ALTER TABLE DISTRICT38 DROP CONSTRAINT
DISTRICT38CKC;
ALTER TABLE DISTRICT38 ADD CONSTRAINT
DISTRICT38CKC CHECK (D_W_ID BETWEEN 59201 AND
60800);
SET INTEGRITY FOR DISTRICT38 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT39 OFF;
ALTER TABLE DISTRICT39 DROP CONSTRAINT
DISTRICT39CKC;
ALTER TABLE DISTRICT39 ADD CONSTRAINT
DISTRICT39CKC CHECK (D_W_ID BETWEEN 60801 AND
62400);
SET INTEGRITY FOR DISTRICT39 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT40 OFF;
ALTER TABLE DISTRICT40 DROP CONSTRAINT
DISTRICT40CKC;
ALTER TABLE DISTRICT40 ADD CONSTRAINT
DISTRICT40CKC CHECK (D_W_ID BETWEEN 62401 AND
64000);
SET INTEGRITY FOR DISTRICT40 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT41 OFF;
ALTER TABLE DISTRICT41 DROP CONSTRAINT
DISTRICT41CKC;
ALTER TABLE DISTRICT41 ADD CONSTRAINT
DISTRICT41CKC CHECK (D_W_ID BETWEEN 64001 AND
65600);
SET INTEGRITY FOR DISTRICT41 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT42 OFF;
ALTER TABLE DISTRICT42 DROP CONSTRAINT
DISTRICT42CKC;
ALTER TABLE DISTRICT42 ADD CONSTRAINT
DISTRICT42CKC CHECK (D_W_ID BETWEEN 65601 AND
67200);
SET INTEGRITY FOR DISTRICT42 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT43 OFF;
ALTER TABLE DISTRICT43 DROP CONSTRAINT
DISTRICT43CKC;

```

```

ALTER TABLE DISTRICT43 ADD CONSTRAINT
DISTRICT43CKC CHECK (D_W_ID BETWEEN 67201 AND
68800);
SET INTEGRITY FOR DISTRICT43 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT44 OFF;
ALTER TABLE DISTRICT44 DROP CONSTRAINT
DISTRICT44CKC;
ALTER TABLE DISTRICT44 ADD CONSTRAINT
DISTRICT44CKC CHECK (D_W_ID BETWEEN 68801 AND
70400);
SET INTEGRITY FOR DISTRICT44 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT45 OFF;
ALTER TABLE DISTRICT45 DROP CONSTRAINT
DISTRICT45CKC;
ALTER TABLE DISTRICT45 ADD CONSTRAINT
DISTRICT45CKC CHECK (D_W_ID BETWEEN 70401 AND
72000);
SET INTEGRITY FOR DISTRICT45 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT46 OFF;
ALTER TABLE DISTRICT46 DROP CONSTRAINT
DISTRICT46CKC;
ALTER TABLE DISTRICT46 ADD CONSTRAINT
DISTRICT46CKC CHECK (D_W_ID BETWEEN 72001 AND
73600);
SET INTEGRITY FOR DISTRICT46 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT47 OFF;
ALTER TABLE DISTRICT47 DROP CONSTRAINT
DISTRICT47CKC;
ALTER TABLE DISTRICT47 ADD CONSTRAINT
DISTRICT47CKC CHECK (D_W_ID BETWEEN 73601 AND
75200);
SET INTEGRITY FOR DISTRICT47 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT48 OFF;
ALTER TABLE DISTRICT48 DROP CONSTRAINT
DISTRICT48CKC;
ALTER TABLE DISTRICT48 ADD CONSTRAINT
DISTRICT48CKC CHECK (D_W_ID BETWEEN 75201 AND
76800);
SET INTEGRITY FOR DISTRICT48 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT49 OFF;
ALTER TABLE DISTRICT49 DROP CONSTRAINT
DISTRICT49CKC;

```

```

ALTER TABLE DISTRICT49 ADD CONSTRAINT
DISTRICT49CKC CHECK (D_W_ID BETWEEN 76801 AND
78400);
SET INTEGRITY FOR DISTRICT49 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT50 OFF;
ALTER TABLE DISTRICT50 DROP CONSTRAINT
DISTRICT50CKC;
ALTER TABLE DISTRICT50 ADD CONSTRAINT
DISTRICT50CKC CHECK (D_W_ID BETWEEN 78401 AND
80000);
SET INTEGRITY FOR DISTRICT50 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT51 OFF;
ALTER TABLE DISTRICT51 DROP CONSTRAINT
DISTRICT51CKC;
ALTER TABLE DISTRICT51 ADD CONSTRAINT
DISTRICT51CKC CHECK (D_W_ID >= 80001);
SET INTEGRITY FOR DISTRICT51 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

DDL/CRCONST HISTORY.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY1 OFF;
ALTER TABLE HISTORY1 DROP CONSTRAINT
HISTORY1CKC;
ALTER TABLE HISTORY1 ADD CONSTRAINT
HISTORY1CKC CHECK (H_W_ID BETWEEN 1 AND 1600);
SET INTEGRITY FOR HISTORY1 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY2 OFF;
ALTER TABLE HISTORY2 DROP CONSTRAINT
HISTORY2CKC;
ALTER TABLE HISTORY2 ADD CONSTRAINT
HISTORY2CKC CHECK (H_W_ID BETWEEN 1601 AND
3200);
SET INTEGRITY FOR HISTORY2 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY3 OFF;
ALTER TABLE HISTORY3 DROP CONSTRAINT
HISTORY3CKC;
ALTER TABLE HISTORY3 ADD CONSTRAINT
HISTORY3CKC CHECK (H_W_ID BETWEEN 3201 AND
4800);
SET INTEGRITY FOR HISTORY3 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY4 OFF;
ALTER TABLE HISTORY4 DROP CONSTRAINT
HISTORY4CKC;

```

```

ALTER TABLE HISTORY4 ADD CONSTRAINT
HISTORY4CKC CHECK (H_W_ID BETWEEN 4801 AND
6400);
SET INTEGRITY FOR HISTORY4 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY5 OFF;
ALTER TABLE HISTORY5 DROP CONSTRAINT
HISTORY5CKC;
ALTER TABLE HISTORY5 ADD CONSTRAINT
HISTORY5CKC CHECK (H_W_ID BETWEEN 6401 AND
8000);
SET INTEGRITY FOR HISTORY5 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY6 OFF;
ALTER TABLE HISTORY6 DROP CONSTRAINT
HISTORY6CKC;
ALTER TABLE HISTORY6 ADD CONSTRAINT
HISTORY6CKC CHECK (H_W_ID BETWEEN 8001 AND
9600);
SET INTEGRITY FOR HISTORY6 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY7 OFF;
ALTER TABLE HISTORY7 DROP CONSTRAINT
HISTORY7CKC;
ALTER TABLE HISTORY7 ADD CONSTRAINT
HISTORY7CKC CHECK (H_W_ID BETWEEN 9601 AND
11200);
SET INTEGRITY FOR HISTORY7 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY8 OFF;
ALTER TABLE HISTORY8 DROP CONSTRAINT
HISTORY8CKC;
ALTER TABLE HISTORY8 ADD CONSTRAINT
HISTORY8CKC CHECK (H_W_ID BETWEEN 11201 AND
12800);
SET INTEGRITY FOR HISTORY8 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY9 OFF;
ALTER TABLE HISTORY9 DROP CONSTRAINT
HISTORY9CKC;
ALTER TABLE HISTORY9 ADD CONSTRAINT
HISTORY9CKC CHECK (H_W_ID BETWEEN 12801 AND
14400);
SET INTEGRITY FOR HISTORY9 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY10 OFF;
ALTER TABLE HISTORY10 DROP CONSTRAINT
HISTORY10CKC;

```

```

ALTER TABLE HISTORY10 ADD CONSTRAINT
HISTORY10CKC CHECK (H_W_ID BETWEEN 14401 AND
16000);
SET INTEGRITY FOR HISTORY10 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY11 OFF;
ALTER TABLE HISTORY11 DROP CONSTRAINT
HISTORY11CKC;
ALTER TABLE HISTORY11 ADD CONSTRAINT
HISTORY11CKC CHECK (H_W_ID BETWEEN 16001 AND
17600);
SET INTEGRITY FOR HISTORY11 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY12 OFF;
ALTER TABLE HISTORY12 DROP CONSTRAINT
HISTORY12CKC;
ALTER TABLE HISTORY12 ADD CONSTRAINT
HISTORY12CKC CHECK (H_W_ID BETWEEN 17601 AND
19200);
SET INTEGRITY FOR HISTORY12 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY13 OFF;
ALTER TABLE HISTORY13 DROP CONSTRAINT
HISTORY13CKC;
ALTER TABLE HISTORY13 ADD CONSTRAINT
HISTORY13CKC CHECK (H_W_ID BETWEEN 19201 AND
20800);
SET INTEGRITY FOR HISTORY13 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY14 OFF;
ALTER TABLE HISTORY14 DROP CONSTRAINT
HISTORY14CKC;
ALTER TABLE HISTORY14 ADD CONSTRAINT
HISTORY14CKC CHECK (H_W_ID BETWEEN 20801 AND
22400);
SET INTEGRITY FOR HISTORY14 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY15 OFF;
ALTER TABLE HISTORY15 DROP CONSTRAINT
HISTORY15CKC;
ALTER TABLE HISTORY15 ADD CONSTRAINT
HISTORY15CKC CHECK (H_W_ID BETWEEN 22401 AND
24000);
SET INTEGRITY FOR HISTORY15 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY16 OFF;
ALTER TABLE HISTORY16 DROP CONSTRAINT
HISTORY16CKC;

```

```

ALTER TABLE HISTORY16 ADD CONSTRAINT
HISTORY16CKC CHECK (H_W_ID BETWEEN 24001 AND
25600);
SET INTEGRITY FOR HISTORY16 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY17 OFF;
ALTER TABLE HISTORY17 DROP CONSTRAINT
HISTORY17CKC;
ALTER TABLE HISTORY17 ADD CONSTRAINT
HISTORY17CKC CHECK (H_W_ID BETWEEN 25601 AND
27200);
SET INTEGRITY FOR HISTORY17 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY18 OFF;
ALTER TABLE HISTORY18 DROP CONSTRAINT
HISTORY18CKC;
ALTER TABLE HISTORY18 ADD CONSTRAINT
HISTORY18CKC CHECK (H_W_ID BETWEEN 27201 AND
28800);
SET INTEGRITY FOR HISTORY18 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY19 OFF;
ALTER TABLE HISTORY19 DROP CONSTRAINT
HISTORY19CKC;
ALTER TABLE HISTORY19 ADD CONSTRAINT
HISTORY19CKC CHECK (H_W_ID BETWEEN 28801 AND
30400);
SET INTEGRITY FOR HISTORY19 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY20 OFF;
ALTER TABLE HISTORY20 DROP CONSTRAINT
HISTORY20CKC;
ALTER TABLE HISTORY20 ADD CONSTRAINT
HISTORY20CKC CHECK (H_W_ID BETWEEN 30401 AND
32000);
SET INTEGRITY FOR HISTORY20 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY21 OFF;
ALTER TABLE HISTORY21 DROP CONSTRAINT
HISTORY21CKC;
ALTER TABLE HISTORY21 ADD CONSTRAINT
HISTORY21CKC CHECK (H_W_ID BETWEEN 32001 AND
33600);
SET INTEGRITY FOR HISTORY21 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY22 OFF;
ALTER TABLE HISTORY22 DROP CONSTRAINT
HISTORY22CKC;

```

```

ALTER TABLE HISTORY22 ADD CONSTRAINT
HISTORY22CKC CHECK (H_W_ID BETWEEN 33601 AND
35200);
SET INTEGRITY FOR HISTORY22 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY23 OFF;
ALTER TABLE HISTORY23 DROP CONSTRAINT
HISTORY23CKC;
ALTER TABLE HISTORY23 ADD CONSTRAINT
HISTORY23CKC CHECK (H_W_ID BETWEEN 35201 AND
36800);
SET INTEGRITY FOR HISTORY23 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY24 OFF;
ALTER TABLE HISTORY24 DROP CONSTRAINT
HISTORY24CKC;
ALTER TABLE HISTORY24 ADD CONSTRAINT
HISTORY24CKC CHECK (H_W_ID BETWEEN 36801 AND
38400);
SET INTEGRITY FOR HISTORY24 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY25 OFF;
ALTER TABLE HISTORY25 DROP CONSTRAINT
HISTORY25CKC;
ALTER TABLE HISTORY25 ADD CONSTRAINT
HISTORY25CKC CHECK (H_W_ID BETWEEN 38401 AND
40000);
SET INTEGRITY FOR HISTORY25 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY26 OFF;
ALTER TABLE HISTORY26 DROP CONSTRAINT
HISTORY26CKC;
ALTER TABLE HISTORY26 ADD CONSTRAINT
HISTORY26CKC CHECK (H_W_ID BETWEEN 40001 AND
41600);
SET INTEGRITY FOR HISTORY26 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY27 OFF;
ALTER TABLE HISTORY27 DROP CONSTRAINT
HISTORY27CKC;
ALTER TABLE HISTORY27 ADD CONSTRAINT
HISTORY27CKC CHECK (H_W_ID BETWEEN 41601 AND
43200);
SET INTEGRITY FOR HISTORY27 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY28 OFF;
ALTER TABLE HISTORY28 DROP CONSTRAINT
HISTORY28CKC;

```

```

ALTER TABLE HISTORY28 ADD CONSTRAINT
HISTORY28CKC CHECK (H_W_ID BETWEEN 43201 AND
44800);
SET INTEGRITY FOR HISTORY28 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY29 OFF;
ALTER TABLE HISTORY29 DROP CONSTRAINT
HISTORY29CKC;
ALTER TABLE HISTORY29 ADD CONSTRAINT
HISTORY29CKC CHECK (H_W_ID BETWEEN 44801 AND
46400);
SET INTEGRITY FOR HISTORY29 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY30 OFF;
ALTER TABLE HISTORY30 DROP CONSTRAINT
HISTORY30CKC;
ALTER TABLE HISTORY30 ADD CONSTRAINT
HISTORY30CKC CHECK (H_W_ID BETWEEN 46401 AND
48000);
SET INTEGRITY FOR HISTORY30 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY31 OFF;
ALTER TABLE HISTORY31 DROP CONSTRAINT
HISTORY31CKC;
ALTER TABLE HISTORY31 ADD CONSTRAINT
HISTORY31CKC CHECK (H_W_ID BETWEEN 48001 AND
49600);
SET INTEGRITY FOR HISTORY31 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY32 OFF;
ALTER TABLE HISTORY32 DROP CONSTRAINT
HISTORY32CKC;
ALTER TABLE HISTORY32 ADD CONSTRAINT
HISTORY32CKC CHECK (H_W_ID BETWEEN 49601 AND
51200);
SET INTEGRITY FOR HISTORY32 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY33 OFF;
ALTER TABLE HISTORY33 DROP CONSTRAINT
HISTORY33CKC;
ALTER TABLE HISTORY33 ADD CONSTRAINT
HISTORY33CKC CHECK (H_W_ID BETWEEN 51201 AND
52800);
SET INTEGRITY FOR HISTORY33 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY34 OFF;
ALTER TABLE HISTORY34 DROP CONSTRAINT
HISTORY34CKC;

```

```

ALTER TABLE HISTORY34 ADD CONSTRAINT
HISTORY34CKC CHECK (H_W_ID BETWEEN 52801 AND
54400);
SET INTEGRITY FOR HISTORY34 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY35 OFF;
ALTER TABLE HISTORY35 DROP CONSTRAINT
HISTORY35CKC;
ALTER TABLE HISTORY35 ADD CONSTRAINT
HISTORY35CKC CHECK (H_W_ID BETWEEN 54401 AND
56000);
SET INTEGRITY FOR HISTORY35 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY36 OFF;
ALTER TABLE HISTORY36 DROP CONSTRAINT
HISTORY36CKC;
ALTER TABLE HISTORY36 ADD CONSTRAINT
HISTORY36CKC CHECK (H_W_ID BETWEEN 56001 AND
57600);
SET INTEGRITY FOR HISTORY36 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY37 OFF;
ALTER TABLE HISTORY37 DROP CONSTRAINT
HISTORY37CKC;
ALTER TABLE HISTORY37 ADD CONSTRAINT
HISTORY37CKC CHECK (H_W_ID BETWEEN 57601 AND
59200);
SET INTEGRITY FOR HISTORY37 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY38 OFF;
ALTER TABLE HISTORY38 DROP CONSTRAINT
HISTORY38CKC;
ALTER TABLE HISTORY38 ADD CONSTRAINT
HISTORY38CKC CHECK (H_W_ID BETWEEN 59201 AND
60800);
SET INTEGRITY FOR HISTORY38 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY39 OFF;
ALTER TABLE HISTORY39 DROP CONSTRAINT
HISTORY39CKC;
ALTER TABLE HISTORY39 ADD CONSTRAINT
HISTORY39CKC CHECK (H_W_ID BETWEEN 60801 AND
62400);
SET INTEGRITY FOR HISTORY39 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY40 OFF;
ALTER TABLE HISTORY40 DROP CONSTRAINT
HISTORY40CKC;

```

```

ALTER TABLE HISTORY40 ADD CONSTRAINT
HISTORY40CKC CHECK (H_W_ID BETWEEN 62401 AND
64000);
SET INTEGRITY FOR HISTORY40 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY41 OFF;
ALTER TABLE HISTORY41 DROP CONSTRAINT
HISTORY41CKC;
ALTER TABLE HISTORY41 ADD CONSTRAINT
HISTORY41CKC CHECK (H_W_ID BETWEEN 64001 AND
65600);
SET INTEGRITY FOR HISTORY41 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY42 OFF;
ALTER TABLE HISTORY42 DROP CONSTRAINT
HISTORY42CKC;
ALTER TABLE HISTORY42 ADD CONSTRAINT
HISTORY42CKC CHECK (H_W_ID BETWEEN 65601 AND
67200);
SET INTEGRITY FOR HISTORY42 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY43 OFF;
ALTER TABLE HISTORY43 DROP CONSTRAINT
HISTORY43CKC;
ALTER TABLE HISTORY43 ADD CONSTRAINT
HISTORY43CKC CHECK (H_W_ID BETWEEN 67201 AND
68800);
SET INTEGRITY FOR HISTORY43 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY44 OFF;
ALTER TABLE HISTORY44 DROP CONSTRAINT
HISTORY44CKC;
ALTER TABLE HISTORY44 ADD CONSTRAINT
HISTORY44CKC CHECK (H_W_ID BETWEEN 68801 AND
70400);
SET INTEGRITY FOR HISTORY44 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY45 OFF;
ALTER TABLE HISTORY45 DROP CONSTRAINT
HISTORY45CKC;
ALTER TABLE HISTORY45 ADD CONSTRAINT
HISTORY45CKC CHECK (H_W_ID BETWEEN 70401 AND
72000);
SET INTEGRITY FOR HISTORY45 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY46 OFF;
ALTER TABLE HISTORY46 DROP CONSTRAINT
HISTORY46CKC;

```

```

ALTER TABLE HISTORY46 ADD CONSTRAINT
HISTORY46CKC CHECK (H_W_ID BETWEEN 72001 AND
73600);
SET INTEGRITY FOR HISTORY46 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY47 OFF;
ALTER TABLE HISTORY47 DROP CONSTRAINT
HISTORY47CKC;
ALTER TABLE HISTORY47 ADD CONSTRAINT
HISTORY47CKC CHECK (H_W_ID BETWEEN 73601 AND
75200);
SET INTEGRITY FOR HISTORY47 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY48 OFF;
ALTER TABLE HISTORY48 DROP CONSTRAINT
HISTORY48CKC;
ALTER TABLE HISTORY48 ADD CONSTRAINT
HISTORY48CKC CHECK (H_W_ID BETWEEN 75201 AND
76800);
SET INTEGRITY FOR HISTORY48 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY49 OFF;
ALTER TABLE HISTORY49 DROP CONSTRAINT
HISTORY49CKC;
ALTER TABLE HISTORY49 ADD CONSTRAINT
HISTORY49CKC CHECK (H_W_ID BETWEEN 76801 AND
78400);
SET INTEGRITY FOR HISTORY49 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY50 OFF;
ALTER TABLE HISTORY50 DROP CONSTRAINT
HISTORY50CKC;
ALTER TABLE HISTORY50 ADD CONSTRAINT
HISTORY50CKC CHECK (H_W_ID BETWEEN 78401 AND
80000);
SET INTEGRITY FOR HISTORY50 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY51 OFF;
ALTER TABLE HISTORY51 DROP CONSTRAINT
HISTORY51CKC;
ALTER TABLE HISTORY51 ADD CONSTRAINT
HISTORY51CKC CHECK (H_W_ID >= 80001);
SET INTEGRITY FOR HISTORY51 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

DDL/CRCONST NEW ORDERA.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA1 OFF;

```

```

ALTER TABLE NEW_ORDERA1 DROP CONSTRAINT
NEW_ORDERA1CKC;
ALTER TABLE NEW_ORDERA1 ADD CONSTRAINT
NEW_ORDERA1CKC CHECK ((NO_W_ID BETWEEN 1 AND
1600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA1 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA2 OFF;
ALTER TABLE NEW_ORDERA2 DROP CONSTRAINT
NEW_ORDERA2CKC;
ALTER TABLE NEW_ORDERA2 ADD CONSTRAINT
NEW_ORDERA2CKC CHECK ((NO_W_ID BETWEEN 1601
AND 3200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA2 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA3 OFF;
ALTER TABLE NEW_ORDERA3 DROP CONSTRAINT
NEW_ORDERA3CKC;
ALTER TABLE NEW_ORDERA3 ADD CONSTRAINT
NEW_ORDERA3CKC CHECK ((NO_W_ID BETWEEN 3201
AND 4800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA3 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA4 OFF;
ALTER TABLE NEW_ORDERA4 DROP CONSTRAINT
NEW_ORDERA4CKC;
ALTER TABLE NEW_ORDERA4 ADD CONSTRAINT
NEW_ORDERA4CKC CHECK ((NO_W_ID BETWEEN 4801
AND 6400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA4 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA5 OFF;
ALTER TABLE NEW_ORDERA5 DROP CONSTRAINT
NEW_ORDERA5CKC;
ALTER TABLE NEW_ORDERA5 ADD CONSTRAINT
NEW_ORDERA5CKC CHECK ((NO_W_ID BETWEEN 6401
AND 8000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA5 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA6 OFF;
ALTER TABLE NEW_ORDERA6 DROP CONSTRAINT
NEW_ORDERA6CKC;
ALTER TABLE NEW_ORDERA6 ADD CONSTRAINT
NEW_ORDERA6CKC CHECK ((NO_W_ID BETWEEN 8001
AND 9600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA6 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA7 OFF;
ALTER TABLE NEW_ORDERA7 DROP CONSTRAINT
NEW_ORDERA7CKC;

```



```

ALTER TABLE NEW_ORDERA7 ADD CONSTRAINT
NEW_ORDERA7CKC CHECK ((NO_W_ID BETWEEN 9601
AND 11200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA7 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA8 OFF;
ALTER TABLE NEW_ORDERA8 DROP CONSTRAINT
NEW_ORDERA8CKC;
ALTER TABLE NEW_ORDERA8 ADD CONSTRAINT
NEW_ORDERA8CKC CHECK ((NO_W_ID BETWEEN 11201
AND 12800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA8 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA9 OFF;
ALTER TABLE NEW_ORDERA9 DROP CONSTRAINT
NEW_ORDERA9CKC;
ALTER TABLE NEW_ORDERA9 ADD CONSTRAINT
NEW_ORDERA9CKC CHECK ((NO_W_ID BETWEEN 12801
AND 14400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA9 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA10 OFF;
ALTER TABLE NEW_ORDERA10 DROP CONSTRAINT
NEW_ORDERA10CKC;
ALTER TABLE NEW_ORDERA10 ADD CONSTRAINT
NEW_ORDERA10CKC CHECK ((NO_W_ID BETWEEN 14401
AND 16000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA10 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA11 OFF;
ALTER TABLE NEW_ORDERA11 DROP CONSTRAINT
NEW_ORDERA11CKC;
ALTER TABLE NEW_ORDERA11 ADD CONSTRAINT
NEW_ORDERA11CKC CHECK ((NO_W_ID BETWEEN 16001
AND 17600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA11 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA12 OFF;
ALTER TABLE NEW_ORDERA12 DROP CONSTRAINT
NEW_ORDERA12CKC;
ALTER TABLE NEW_ORDERA12 ADD CONSTRAINT
NEW_ORDERA12CKC CHECK ((NO_W_ID BETWEEN 17601
AND 19200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA12 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA13 OFF;
ALTER TABLE NEW_ORDERA13 DROP CONSTRAINT
NEW_ORDERA13CKC;

```

```

ALTER TABLE NEW_ORDERA13 ADD CONSTRAINT
NEW_ORDERA13CKC CHECK ((NO_W_ID BETWEEN 19201
AND 20800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA13 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA14 OFF;
ALTER TABLE NEW_ORDERA14 DROP CONSTRAINT
NEW_ORDERA14CKC;
ALTER TABLE NEW_ORDERA14 ADD CONSTRAINT
NEW_ORDERA14CKC CHECK ((NO_W_ID BETWEEN 20801
AND 22400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA14 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA15 OFF;
ALTER TABLE NEW_ORDERA15 DROP CONSTRAINT
NEW_ORDERA15CKC;
ALTER TABLE NEW_ORDERA15 ADD CONSTRAINT
NEW_ORDERA15CKC CHECK ((NO_W_ID BETWEEN 22401
AND 24000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA15 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA16 OFF;
ALTER TABLE NEW_ORDERA16 DROP CONSTRAINT
NEW_ORDERA16CKC;
ALTER TABLE NEW_ORDERA16 ADD CONSTRAINT
NEW_ORDERA16CKC CHECK ((NO_W_ID BETWEEN 24001
AND 25600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA16 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA17 OFF;
ALTER TABLE NEW_ORDERA17 DROP CONSTRAINT
NEW_ORDERA17CKC;
ALTER TABLE NEW_ORDERA17 ADD CONSTRAINT
NEW_ORDERA17CKC CHECK ((NO_W_ID BETWEEN 25601
AND 27200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA17 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA18 OFF;
ALTER TABLE NEW_ORDERA18 DROP CONSTRAINT
NEW_ORDERA18CKC;
ALTER TABLE NEW_ORDERA18 ADD CONSTRAINT
NEW_ORDERA18CKC CHECK ((NO_W_ID BETWEEN 27201
AND 28800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA18 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA19 OFF;
ALTER TABLE NEW_ORDERA19 DROP CONSTRAINT
NEW_ORDERA19CKC;

```

```

ALTER TABLE NEW_ORDERA19 ADD CONSTRAINT
NEW_ORDERA19CKC CHECK ((NO_W_ID BETWEEN 28801
AND 30400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA19 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA20 OFF;
ALTER TABLE NEW_ORDERA20 DROP CONSTRAINT
NEW_ORDERA20CKC;
ALTER TABLE NEW_ORDERA20 ADD CONSTRAINT
NEW_ORDERA20CKC CHECK ((NO_W_ID BETWEEN 30401
AND 32000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA20 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA21 OFF;
ALTER TABLE NEW_ORDERA21 DROP CONSTRAINT
NEW_ORDERA21CKC;
ALTER TABLE NEW_ORDERA21 ADD CONSTRAINT
NEW_ORDERA21CKC CHECK ((NO_W_ID BETWEEN 32001
AND 33600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA21 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA22 OFF;
ALTER TABLE NEW_ORDERA22 DROP CONSTRAINT
NEW_ORDERA22CKC;
ALTER TABLE NEW_ORDERA22 ADD CONSTRAINT
NEW_ORDERA22CKC CHECK ((NO_W_ID BETWEEN 33601
AND 35200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA22 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA23 OFF;
ALTER TABLE NEW_ORDERA23 DROP CONSTRAINT
NEW_ORDERA23CKC;
ALTER TABLE NEW_ORDERA23 ADD CONSTRAINT
NEW_ORDERA23CKC CHECK ((NO_W_ID BETWEEN 35201
AND 36800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA23 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA24 OFF;
ALTER TABLE NEW_ORDERA24 DROP CONSTRAINT
NEW_ORDERA24CKC;
ALTER TABLE NEW_ORDERA24 ADD CONSTRAINT
NEW_ORDERA24CKC CHECK ((NO_W_ID BETWEEN 36801
AND 38400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA24 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA25 OFF;
ALTER TABLE NEW_ORDERA25 DROP CONSTRAINT
NEW_ORDERA25CKC;

```

```

ALTER TABLE NEW_ORDERA25 ADD CONSTRAINT
NEW_ORDERA25CKC CHECK ((NO_W_ID BETWEEN 38401
AND 40000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA25 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA26 OFF;
ALTER TABLE NEW_ORDERA26 DROP CONSTRAINT
NEW_ORDERA26CKC;
ALTER TABLE NEW_ORDERA26 ADD CONSTRAINT
NEW_ORDERA26CKC CHECK ((NO_W_ID BETWEEN 40001
AND 41600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA26 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA27 OFF;
ALTER TABLE NEW_ORDERA27 DROP CONSTRAINT
NEW_ORDERA27CKC;
ALTER TABLE NEW_ORDERA27 ADD CONSTRAINT
NEW_ORDERA27CKC CHECK ((NO_W_ID BETWEEN 41601
AND 43200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA27 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA28 OFF;
ALTER TABLE NEW_ORDERA28 DROP CONSTRAINT
NEW_ORDERA28CKC;
ALTER TABLE NEW_ORDERA28 ADD CONSTRAINT
NEW_ORDERA28CKC CHECK ((NO_W_ID BETWEEN 43201
AND 44800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA28 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA29 OFF;
ALTER TABLE NEW_ORDERA29 DROP CONSTRAINT
NEW_ORDERA29CKC;
ALTER TABLE NEW_ORDERA29 ADD CONSTRAINT
NEW_ORDERA29CKC CHECK ((NO_W_ID BETWEEN 44801
AND 46400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA29 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA30 OFF;
ALTER TABLE NEW_ORDERA30 DROP CONSTRAINT
NEW_ORDERA30CKC;
ALTER TABLE NEW_ORDERA30 ADD CONSTRAINT
NEW_ORDERA30CKC CHECK ((NO_W_ID BETWEEN 46401
AND 48000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA30 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA31 OFF;
ALTER TABLE NEW_ORDERA31 DROP CONSTRAINT
NEW_ORDERA31CKC;

```

```

ALTER TABLE NEW_ORDERA31 ADD CONSTRAINT
NEW_ORDERA31CKC CHECK ((NO_W_ID BETWEEN 48001
AND 49600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA31 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA32 OFF;
ALTER TABLE NEW_ORDERA32 DROP CONSTRAINT
NEW_ORDERA32CKC;
ALTER TABLE NEW_ORDERA32 ADD CONSTRAINT
NEW_ORDERA32CKC CHECK ((NO_W_ID BETWEEN 49601
AND 51200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA32 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA33 OFF;
ALTER TABLE NEW_ORDERA33 DROP CONSTRAINT
NEW_ORDERA33CKC;
ALTER TABLE NEW_ORDERA33 ADD CONSTRAINT
NEW_ORDERA33CKC CHECK ((NO_W_ID BETWEEN 51201
AND 52800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA33 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA34 OFF;
ALTER TABLE NEW_ORDERA34 DROP CONSTRAINT
NEW_ORDERA34CKC;
ALTER TABLE NEW_ORDERA34 ADD CONSTRAINT
NEW_ORDERA34CKC CHECK ((NO_W_ID BETWEEN 52801
AND 54400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA34 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA35 OFF;
ALTER TABLE NEW_ORDERA35 DROP CONSTRAINT
NEW_ORDERA35CKC;
ALTER TABLE NEW_ORDERA35 ADD CONSTRAINT
NEW_ORDERA35CKC CHECK ((NO_W_ID BETWEEN 54401
AND 56000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA35 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA36 OFF;
ALTER TABLE NEW_ORDERA36 DROP CONSTRAINT
NEW_ORDERA36CKC;
ALTER TABLE NEW_ORDERA36 ADD CONSTRAINT
NEW_ORDERA36CKC CHECK ((NO_W_ID BETWEEN 56001
AND 57600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA36 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA37 OFF;
ALTER TABLE NEW_ORDERA37 DROP CONSTRAINT
NEW_ORDERA37CKC;

```

```

ALTER TABLE NEW_ORDERA37 ADD CONSTRAINT
NEW_ORDERA37CKC CHECK ((NO_W_ID BETWEEN 57601
AND 59200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA37 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA38 OFF;
ALTER TABLE NEW_ORDERA38 DROP CONSTRAINT
NEW_ORDERA38CKC;
ALTER TABLE NEW_ORDERA38 ADD CONSTRAINT
NEW_ORDERA38CKC CHECK ((NO_W_ID BETWEEN 59201
AND 60800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA38 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA39 OFF;
ALTER TABLE NEW_ORDERA39 DROP CONSTRAINT
NEW_ORDERA39CKC;
ALTER TABLE NEW_ORDERA39 ADD CONSTRAINT
NEW_ORDERA39CKC CHECK ((NO_W_ID BETWEEN 60801
AND 62400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA39 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA40 OFF;
ALTER TABLE NEW_ORDERA40 DROP CONSTRAINT
NEW_ORDERA40CKC;
ALTER TABLE NEW_ORDERA40 ADD CONSTRAINT
NEW_ORDERA40CKC CHECK ((NO_W_ID BETWEEN 62401
AND 64000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA40 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA41 OFF;
ALTER TABLE NEW_ORDERA41 DROP CONSTRAINT
NEW_ORDERA41CKC;
ALTER TABLE NEW_ORDERA41 ADD CONSTRAINT
NEW_ORDERA41CKC CHECK ((NO_W_ID BETWEEN 64001
AND 65600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA41 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA42 OFF;
ALTER TABLE NEW_ORDERA42 DROP CONSTRAINT
NEW_ORDERA42CKC;
ALTER TABLE NEW_ORDERA42 ADD CONSTRAINT
NEW_ORDERA42CKC CHECK ((NO_W_ID BETWEEN 65601
AND 67200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA42 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA43 OFF;
ALTER TABLE NEW_ORDERA43 DROP CONSTRAINT
NEW_ORDERA43CKC;

```

```

ALTER TABLE NEW_ORDERA43 ADD CONSTRAINT
NEW_ORDERA43CKC CHECK ((NO_W_ID BETWEEN 67201
AND 68800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA43 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA44 OFF;
ALTER TABLE NEW_ORDERA44 DROP CONSTRAINT
NEW_ORDERA44CKC;
ALTER TABLE NEW_ORDERA44 ADD CONSTRAINT
NEW_ORDERA44CKC CHECK ((NO_W_ID BETWEEN 68801
AND 70400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA44 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA45 OFF;
ALTER TABLE NEW_ORDERA45 DROP CONSTRAINT
NEW_ORDERA45CKC;
ALTER TABLE NEW_ORDERA45 ADD CONSTRAINT
NEW_ORDERA45CKC CHECK ((NO_W_ID BETWEEN 70401
AND 72000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA45 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA46 OFF;
ALTER TABLE NEW_ORDERA46 DROP CONSTRAINT
NEW_ORDERA46CKC;
ALTER TABLE NEW_ORDERA46 ADD CONSTRAINT
NEW_ORDERA46CKC CHECK ((NO_W_ID BETWEEN 72001
AND 73600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA46 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA47 OFF;
ALTER TABLE NEW_ORDERA47 DROP CONSTRAINT
NEW_ORDERA47CKC;
ALTER TABLE NEW_ORDERA47 ADD CONSTRAINT
NEW_ORDERA47CKC CHECK ((NO_W_ID BETWEEN 73601
AND 75200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA47 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA48 OFF;
ALTER TABLE NEW_ORDERA48 DROP CONSTRAINT
NEW_ORDERA48CKC;
ALTER TABLE NEW_ORDERA48 ADD CONSTRAINT
NEW_ORDERA48CKC CHECK ((NO_W_ID BETWEEN 75201
AND 76800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA48 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA49 OFF;
ALTER TABLE NEW_ORDERA49 DROP CONSTRAINT
NEW_ORDERA49CKC;

```

```

ALTER TABLE NEW_ORDERA49 ADD CONSTRAINT
NEW_ORDERA49CKC CHECK ((NO_W_ID BETWEEN 76801
AND 78400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA49 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA50 OFF;
ALTER TABLE NEW_ORDERA50 DROP CONSTRAINT
NEW_ORDERA50CKC;
ALTER TABLE NEW_ORDERA50 ADD CONSTRAINT
NEW_ORDERA50CKC CHECK ((NO_W_ID BETWEEN 78401
AND 80000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA50 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA51 OFF;
ALTER TABLE NEW_ORDERA51 DROP CONSTRAINT
NEW_ORDERA51CKC;
ALTER TABLE NEW_ORDERA51 ADD CONSTRAINT
NEW_ORDERA51CKC CHECK ((NO_W_ID >= 80001) AND
(NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA51 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

DDL/CRCONST NEW ORDERB.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB1 OFF;
ALTER TABLE NEW_ORDERB1 DROP CONSTRAINT
NEW_ORDERB1CKC;
ALTER TABLE NEW_ORDERB1 ADD CONSTRAINT
NEW_ORDERB1CKC CHECK ((NO_W_ID BETWEEN 1 AND
1600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB1 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB2 OFF;
ALTER TABLE NEW_ORDERB2 DROP CONSTRAINT
NEW_ORDERB2CKC;
ALTER TABLE NEW_ORDERB2 ADD CONSTRAINT
NEW_ORDERB2CKC CHECK ((NO_W_ID BETWEEN 1601
AND 3200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB2 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB3 OFF;
ALTER TABLE NEW_ORDERB3 DROP CONSTRAINT
NEW_ORDERB3CKC;
ALTER TABLE NEW_ORDERB3 ADD CONSTRAINT
NEW_ORDERB3CKC CHECK ((NO_W_ID BETWEEN 3201
AND 4800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB3 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB4 OFF;

```

```

ALTER TABLE NEW_ORDERB4 DROP CONSTRAINT
NEW_ORDERB4CKC;
ALTER TABLE NEW_ORDERB4 ADD CONSTRAINT
NEW_ORDERB4CKC CHECK ((NO_W_ID BETWEEN 4801
AND 6400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB4 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB5 OFF;
ALTER TABLE NEW_ORDERB5 DROP CONSTRAINT
NEW_ORDERB5CKC;
ALTER TABLE NEW_ORDERB5 ADD CONSTRAINT
NEW_ORDERB5CKC CHECK ((NO_W_ID BETWEEN 6401
AND 8000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB5 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB6 OFF;
ALTER TABLE NEW_ORDERB6 DROP CONSTRAINT
NEW_ORDERB6CKC;
ALTER TABLE NEW_ORDERB6 ADD CONSTRAINT
NEW_ORDERB6CKC CHECK ((NO_W_ID BETWEEN 8001
AND 9600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB6 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB7 OFF;
ALTER TABLE NEW_ORDERB7 DROP CONSTRAINT
NEW_ORDERB7CKC;
ALTER TABLE NEW_ORDERB7 ADD CONSTRAINT
NEW_ORDERB7CKC CHECK ((NO_W_ID BETWEEN 9601
AND 11200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB7 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB8 OFF;
ALTER TABLE NEW_ORDERB8 DROP CONSTRAINT
NEW_ORDERB8CKC;
ALTER TABLE NEW_ORDERB8 ADD CONSTRAINT
NEW_ORDERB8CKC CHECK ((NO_W_ID BETWEEN 11201
AND 12800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB8 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB9 OFF;
ALTER TABLE NEW_ORDERB9 DROP CONSTRAINT
NEW_ORDERB9CKC;
ALTER TABLE NEW_ORDERB9 ADD CONSTRAINT
NEW_ORDERB9CKC CHECK ((NO_W_ID BETWEEN 12801
AND 14400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB9 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB10 OFF;
ALTER TABLE NEW_ORDERB10 DROP CONSTRAINT
NEW_ORDERB10CKC;

```

```

ALTER TABLE NEW_ORDERB10 ADD CONSTRAINT
NEW_ORDERB10CKC CHECK ((NO_W_ID BETWEEN 14401
AND 16000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB10 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB11 OFF;
ALTER TABLE NEW_ORDERB11 DROP CONSTRAINT
NEW_ORDERB11CKC;
ALTER TABLE NEW_ORDERB11 ADD CONSTRAINT
NEW_ORDERB11CKC CHECK ((NO_W_ID BETWEEN 16001
AND 17600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB11 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB12 OFF;
ALTER TABLE NEW_ORDERB12 DROP CONSTRAINT
NEW_ORDERB12CKC;
ALTER TABLE NEW_ORDERB12 ADD CONSTRAINT
NEW_ORDERB12CKC CHECK ((NO_W_ID BETWEEN 17601
AND 19200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB12 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB13 OFF;
ALTER TABLE NEW_ORDERB13 DROP CONSTRAINT
NEW_ORDERB13CKC;
ALTER TABLE NEW_ORDERB13 ADD CONSTRAINT
NEW_ORDERB13CKC CHECK ((NO_W_ID BETWEEN 19201
AND 20800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB13 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB14 OFF;
ALTER TABLE NEW_ORDERB14 DROP CONSTRAINT
NEW_ORDERB14CKC;
ALTER TABLE NEW_ORDERB14 ADD CONSTRAINT
NEW_ORDERB14CKC CHECK ((NO_W_ID BETWEEN 20801
AND 22400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB14 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB15 OFF;
ALTER TABLE NEW_ORDERB15 DROP CONSTRAINT
NEW_ORDERB15CKC;
ALTER TABLE NEW_ORDERB15 ADD CONSTRAINT
NEW_ORDERB15CKC CHECK ((NO_W_ID BETWEEN 22401
AND 24000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB15 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB16 OFF;
ALTER TABLE NEW_ORDERB16 DROP CONSTRAINT
NEW_ORDERB16CKC;

```

```

ALTER TABLE NEW_ORDERB16 ADD CONSTRAINT
NEW_ORDERB16CKC CHECK ((NO_W_ID BETWEEN 24001
AND 25600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB16 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB17 OFF;
ALTER TABLE NEW_ORDERB17 DROP CONSTRAINT
NEW_ORDERB17CKC;
ALTER TABLE NEW_ORDERB17 ADD CONSTRAINT
NEW_ORDERB17CKC CHECK ((NO_W_ID BETWEEN 25601
AND 27200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB17 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB18 OFF;
ALTER TABLE NEW_ORDERB18 DROP CONSTRAINT
NEW_ORDERB18CKC;
ALTER TABLE NEW_ORDERB18 ADD CONSTRAINT
NEW_ORDERB18CKC CHECK ((NO_W_ID BETWEEN 27201
AND 28800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB18 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB19 OFF;
ALTER TABLE NEW_ORDERB19 DROP CONSTRAINT
NEW_ORDERB19CKC;
ALTER TABLE NEW_ORDERB19 ADD CONSTRAINT
NEW_ORDERB19CKC CHECK ((NO_W_ID BETWEEN 28801
AND 30400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB19 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB20 OFF;
ALTER TABLE NEW_ORDERB20 DROP CONSTRAINT
NEW_ORDERB20CKC;
ALTER TABLE NEW_ORDERB20 ADD CONSTRAINT
NEW_ORDERB20CKC CHECK ((NO_W_ID BETWEEN 30401
AND 32000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB20 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB21 OFF;
ALTER TABLE NEW_ORDERB21 DROP CONSTRAINT
NEW_ORDERB21CKC;
ALTER TABLE NEW_ORDERB21 ADD CONSTRAINT
NEW_ORDERB21CKC CHECK ((NO_W_ID BETWEEN 32001
AND 33600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB21 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB22 OFF;
ALTER TABLE NEW_ORDERB22 DROP CONSTRAINT
NEW_ORDERB22CKC;

```

```

ALTER TABLE NEW_ORDERB22 ADD CONSTRAINT
NEW_ORDERB22CKC CHECK ((NO_W_ID BETWEEN 33601
AND 35200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB22 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB23 OFF;
ALTER TABLE NEW_ORDERB23 DROP CONSTRAINT
NEW_ORDERB23CKC;
ALTER TABLE NEW_ORDERB23 ADD CONSTRAINT
NEW_ORDERB23CKC CHECK ((NO_W_ID BETWEEN 35201
AND 36800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB23 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB24 OFF;
ALTER TABLE NEW_ORDERB24 DROP CONSTRAINT
NEW_ORDERB24CKC;
ALTER TABLE NEW_ORDERB24 ADD CONSTRAINT
NEW_ORDERB24CKC CHECK ((NO_W_ID BETWEEN 36801
AND 38400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB24 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB25 OFF;
ALTER TABLE NEW_ORDERB25 DROP CONSTRAINT
NEW_ORDERB25CKC;
ALTER TABLE NEW_ORDERB25 ADD CONSTRAINT
NEW_ORDERB25CKC CHECK ((NO_W_ID BETWEEN 38401
AND 40000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB25 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB26 OFF;
ALTER TABLE NEW_ORDERB26 DROP CONSTRAINT
NEW_ORDERB26CKC;
ALTER TABLE NEW_ORDERB26 ADD CONSTRAINT
NEW_ORDERB26CKC CHECK ((NO_W_ID BETWEEN 40001
AND 41600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB26 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB27 OFF;
ALTER TABLE NEW_ORDERB27 DROP CONSTRAINT
NEW_ORDERB27CKC;
ALTER TABLE NEW_ORDERB27 ADD CONSTRAINT
NEW_ORDERB27CKC CHECK ((NO_W_ID BETWEEN 41601
AND 43200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB27 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB28 OFF;
ALTER TABLE NEW_ORDERB28 DROP CONSTRAINT
NEW_ORDERB28CKC;

```

```

ALTER TABLE NEW_ORDERB28 ADD CONSTRAINT
NEW_ORDERB28CKC CHECK ((NO_W_ID BETWEEN 43201
AND 44800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB28 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB29 OFF;
ALTER TABLE NEW_ORDERB29 DROP CONSTRAINT
NEW_ORDERB29CKC;
ALTER TABLE NEW_ORDERB29 ADD CONSTRAINT
NEW_ORDERB29CKC CHECK ((NO_W_ID BETWEEN 44801
AND 46400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB29 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB30 OFF;
ALTER TABLE NEW_ORDERB30 DROP CONSTRAINT
NEW_ORDERB30CKC;
ALTER TABLE NEW_ORDERB30 ADD CONSTRAINT
NEW_ORDERB30CKC CHECK ((NO_W_ID BETWEEN 46401
AND 48000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB30 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB31 OFF;
ALTER TABLE NEW_ORDERB31 DROP CONSTRAINT
NEW_ORDERB31CKC;
ALTER TABLE NEW_ORDERB31 ADD CONSTRAINT
NEW_ORDERB31CKC CHECK ((NO_W_ID BETWEEN 48001
AND 49600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB31 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB32 OFF;
ALTER TABLE NEW_ORDERB32 DROP CONSTRAINT
NEW_ORDERB32CKC;
ALTER TABLE NEW_ORDERB32 ADD CONSTRAINT
NEW_ORDERB32CKC CHECK ((NO_W_ID BETWEEN 49601
AND 51200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB32 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB33 OFF;
ALTER TABLE NEW_ORDERB33 DROP CONSTRAINT
NEW_ORDERB33CKC;
ALTER TABLE NEW_ORDERB33 ADD CONSTRAINT
NEW_ORDERB33CKC CHECK ((NO_W_ID BETWEEN 51201
AND 52800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB33 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB34 OFF;
ALTER TABLE NEW_ORDERB34 DROP CONSTRAINT
NEW_ORDERB34CKC;

```

```

ALTER TABLE NEW_ORDERB34 ADD CONSTRAINT
NEW_ORDERB34CKC CHECK ((NO_W_ID BETWEEN 52801
AND 54400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB34 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB35 OFF;
ALTER TABLE NEW_ORDERB35 DROP CONSTRAINT
NEW_ORDERB35CKC;
ALTER TABLE NEW_ORDERB35 ADD CONSTRAINT
NEW_ORDERB35CKC CHECK ((NO_W_ID BETWEEN 54401
AND 56000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB35 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB36 OFF;
ALTER TABLE NEW_ORDERB36 DROP CONSTRAINT
NEW_ORDERB36CKC;
ALTER TABLE NEW_ORDERB36 ADD CONSTRAINT
NEW_ORDERB36CKC CHECK ((NO_W_ID BETWEEN 56001
AND 57600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB36 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB37 OFF;
ALTER TABLE NEW_ORDERB37 DROP CONSTRAINT
NEW_ORDERB37CKC;
ALTER TABLE NEW_ORDERB37 ADD CONSTRAINT
NEW_ORDERB37CKC CHECK ((NO_W_ID BETWEEN 57601
AND 59200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB37 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB38 OFF;
ALTER TABLE NEW_ORDERB38 DROP CONSTRAINT
NEW_ORDERB38CKC;
ALTER TABLE NEW_ORDERB38 ADD CONSTRAINT
NEW_ORDERB38CKC CHECK ((NO_W_ID BETWEEN 59201
AND 60800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB38 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB39 OFF;
ALTER TABLE NEW_ORDERB39 DROP CONSTRAINT
NEW_ORDERB39CKC;
ALTER TABLE NEW_ORDERB39 ADD CONSTRAINT
NEW_ORDERB39CKC CHECK ((NO_W_ID BETWEEN 60801
AND 62400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB39 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB40 OFF;
ALTER TABLE NEW_ORDERB40 DROP CONSTRAINT
NEW_ORDERB40CKC;

```

```

ALTER TABLE NEW_ORDERB40 ADD CONSTRAINT
NEW_ORDERB40CKC CHECK ((NO_W_ID BETWEEN 62401
AND 64000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB40 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB41 OFF;
ALTER TABLE NEW_ORDERB41 DROP CONSTRAINT
NEW_ORDERB41CKC;
ALTER TABLE NEW_ORDERB41 ADD CONSTRAINT
NEW_ORDERB41CKC CHECK ((NO_W_ID BETWEEN 64001
AND 65600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB41 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB42 OFF;
ALTER TABLE NEW_ORDERB42 DROP CONSTRAINT
NEW_ORDERB42CKC;
ALTER TABLE NEW_ORDERB42 ADD CONSTRAINT
NEW_ORDERB42CKC CHECK ((NO_W_ID BETWEEN 65601
AND 67200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB42 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB43 OFF;
ALTER TABLE NEW_ORDERB43 DROP CONSTRAINT
NEW_ORDERB43CKC;
ALTER TABLE NEW_ORDERB43 ADD CONSTRAINT
NEW_ORDERB43CKC CHECK ((NO_W_ID BETWEEN 67201
AND 68800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB43 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB44 OFF;
ALTER TABLE NEW_ORDERB44 DROP CONSTRAINT
NEW_ORDERB44CKC;
ALTER TABLE NEW_ORDERB44 ADD CONSTRAINT
NEW_ORDERB44CKC CHECK ((NO_W_ID BETWEEN 68801
AND 70400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB44 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB45 OFF;
ALTER TABLE NEW_ORDERB45 DROP CONSTRAINT
NEW_ORDERB45CKC;
ALTER TABLE NEW_ORDERB45 ADD CONSTRAINT
NEW_ORDERB45CKC CHECK ((NO_W_ID BETWEEN 70401
AND 72000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB45 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB46 OFF;
ALTER TABLE NEW_ORDERB46 DROP CONSTRAINT
NEW_ORDERB46CKC;

```

```

ALTER TABLE NEW_ORDERB46 ADD CONSTRAINT
NEW_ORDERB46CKC CHECK ((NO_W_ID BETWEEN 72001
AND 73600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB46 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB47 OFF;
ALTER TABLE NEW_ORDERB47 DROP CONSTRAINT
NEW_ORDERB47CKC;
ALTER TABLE NEW_ORDERB47 ADD CONSTRAINT
NEW_ORDERB47CKC CHECK ((NO_W_ID BETWEEN 73601
AND 75200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB47 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB48 OFF;
ALTER TABLE NEW_ORDERB48 DROP CONSTRAINT
NEW_ORDERB48CKC;
ALTER TABLE NEW_ORDERB48 ADD CONSTRAINT
NEW_ORDERB48CKC CHECK ((NO_W_ID BETWEEN 75201
AND 76800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB48 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB49 OFF;
ALTER TABLE NEW_ORDERB49 DROP CONSTRAINT
NEW_ORDERB49CKC;
ALTER TABLE NEW_ORDERB49 ADD CONSTRAINT
NEW_ORDERB49CKC CHECK ((NO_W_ID BETWEEN 76801
AND 78400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB49 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB50 OFF;
ALTER TABLE NEW_ORDERB50 DROP CONSTRAINT
NEW_ORDERB50CKC;
ALTER TABLE NEW_ORDERB50 ADD CONSTRAINT
NEW_ORDERB50CKC CHECK ((NO_W_ID BETWEEN 78401
AND 80000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB50 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB51 OFF;
ALTER TABLE NEW_ORDERB51 DROP CONSTRAINT
NEW_ORDERB51CKC;
ALTER TABLE NEW_ORDERB51 ADD CONSTRAINT
NEW_ORDERB51CKC CHECK ((NO_W_ID >= 80001) AND
(NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB51 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

DDL/CRCONST ORDERS.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS1 OFF;

```

```

ALTER TABLE ORDERS1 DROP CONSTRAINT
ORDERS1CKC;
ALTER TABLE ORDERS1 ADD CONSTRAINT ORDERS1CKC
CHECK (O_W_ID BETWEEN 1 AND 1600);
SET INTEGRITY FOR ORDERS1 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS2 OFF;
ALTER TABLE ORDERS2 DROP CONSTRAINT
ORDERS2CKC;
ALTER TABLE ORDERS2 ADD CONSTRAINT ORDERS2CKC
CHECK (O_W_ID BETWEEN 1601 AND 3200);
SET INTEGRITY FOR ORDERS2 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS3 OFF;
ALTER TABLE ORDERS3 DROP CONSTRAINT
ORDERS3CKC;
ALTER TABLE ORDERS3 ADD CONSTRAINT ORDERS3CKC
CHECK (O_W_ID BETWEEN 3201 AND 4800);
SET INTEGRITY FOR ORDERS3 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS4 OFF;
ALTER TABLE ORDERS4 DROP CONSTRAINT
ORDERS4CKC;
ALTER TABLE ORDERS4 ADD CONSTRAINT ORDERS4CKC
CHECK (O_W_ID BETWEEN 4801 AND 6400);
SET INTEGRITY FOR ORDERS4 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS5 OFF;
ALTER TABLE ORDERS5 DROP CONSTRAINT
ORDERS5CKC;
ALTER TABLE ORDERS5 ADD CONSTRAINT ORDERS5CKC
CHECK (O_W_ID BETWEEN 6401 AND 8000);
SET INTEGRITY FOR ORDERS5 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS6 OFF;
ALTER TABLE ORDERS6 DROP CONSTRAINT
ORDERS6CKC;
ALTER TABLE ORDERS6 ADD CONSTRAINT ORDERS6CKC
CHECK (O_W_ID BETWEEN 8001 AND 9600);
SET INTEGRITY FOR ORDERS6 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS7 OFF;
ALTER TABLE ORDERS7 DROP CONSTRAINT
ORDERS7CKC;
ALTER TABLE ORDERS7 ADD CONSTRAINT ORDERS7CKC
CHECK (O_W_ID BETWEEN 9601 AND 11200);
SET INTEGRITY FOR ORDERS7 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;

```

```

SET INTEGRITY FOR ORDERS8 OFF;
ALTER TABLE ORDERS8 DROP CONSTRAINT
ORDERS8CKC;
ALTER TABLE ORDERS8 ADD CONSTRAINT ORDERS8CKC
CHECK (O_W_ID BETWEEN 11201 AND 12800);
SET INTEGRITY FOR ORDERS8 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS9 OFF;
ALTER TABLE ORDERS9 DROP CONSTRAINT
ORDERS9CKC;
ALTER TABLE ORDERS9 ADD CONSTRAINT ORDERS9CKC
CHECK (O_W_ID BETWEEN 12801 AND 14400);
SET INTEGRITY FOR ORDERS9 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS10 OFF;
ALTER TABLE ORDERS10 DROP CONSTRAINT
ORDERS10CKC;
ALTER TABLE ORDERS10 ADD CONSTRAINT
ORDERS10CKC CHECK (O_W_ID BETWEEN 14401 AND
16000);
SET INTEGRITY FOR ORDERS10 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS11 OFF;
ALTER TABLE ORDERS11 DROP CONSTRAINT
ORDERS11CKC;
ALTER TABLE ORDERS11 ADD CONSTRAINT
ORDERS11CKC CHECK (O_W_ID BETWEEN 16001 AND
17600);
SET INTEGRITY FOR ORDERS11 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS12 OFF;
ALTER TABLE ORDERS12 DROP CONSTRAINT
ORDERS12CKC;
ALTER TABLE ORDERS12 ADD CONSTRAINT
ORDERS12CKC CHECK (O_W_ID BETWEEN 17601 AND
19200);
SET INTEGRITY FOR ORDERS12 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS13 OFF;
ALTER TABLE ORDERS13 DROP CONSTRAINT
ORDERS13CKC;
ALTER TABLE ORDERS13 ADD CONSTRAINT
ORDERS13CKC CHECK (O_W_ID BETWEEN 19201 AND
20800);
SET INTEGRITY FOR ORDERS13 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS14 OFF;
ALTER TABLE ORDERS14 DROP CONSTRAINT
ORDERS14CKC;

```

```

ALTER TABLE ORDERS14 ADD CONSTRAINT
ORDERS14CKC CHECK (O_W_ID BETWEEN 20801 AND
22400);
SET INTEGRITY FOR ORDERS14 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS15 OFF;
ALTER TABLE ORDERS15 DROP CONSTRAINT
ORDERS15CKC;
ALTER TABLE ORDERS15 ADD CONSTRAINT
ORDERS15CKC CHECK (O_W_ID BETWEEN 22401 AND
24000);
SET INTEGRITY FOR ORDERS15 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS16 OFF;
ALTER TABLE ORDERS16 DROP CONSTRAINT
ORDERS16CKC;
ALTER TABLE ORDERS16 ADD CONSTRAINT
ORDERS16CKC CHECK (O_W_ID BETWEEN 24001 AND
25600);
SET INTEGRITY FOR ORDERS16 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS17 OFF;
ALTER TABLE ORDERS17 DROP CONSTRAINT
ORDERS17CKC;
ALTER TABLE ORDERS17 ADD CONSTRAINT
ORDERS17CKC CHECK (O_W_ID BETWEEN 25601 AND
27200);
SET INTEGRITY FOR ORDERS17 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS18 OFF;
ALTER TABLE ORDERS18 DROP CONSTRAINT
ORDERS18CKC;
ALTER TABLE ORDERS18 ADD CONSTRAINT
ORDERS18CKC CHECK (O_W_ID BETWEEN 27201 AND
28800);
SET INTEGRITY FOR ORDERS18 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS19 OFF;
ALTER TABLE ORDERS19 DROP CONSTRAINT
ORDERS19CKC;
ALTER TABLE ORDERS19 ADD CONSTRAINT
ORDERS19CKC CHECK (O_W_ID BETWEEN 28801 AND
30400);
SET INTEGRITY FOR ORDERS19 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS20 OFF;
ALTER TABLE ORDERS20 DROP CONSTRAINT
ORDERS20CKC;

```

```

ALTER TABLE ORDERS20 ADD CONSTRAINT
ORDERS20CKC CHECK (O_W_ID BETWEEN 30401 AND
32000);
SET INTEGRITY FOR ORDERS20 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS21 OFF;
ALTER TABLE ORDERS21 DROP CONSTRAINT
ORDERS21CKC;
ALTER TABLE ORDERS21 ADD CONSTRAINT
ORDERS21CKC CHECK (O_W_ID BETWEEN 32001 AND
33600);
SET INTEGRITY FOR ORDERS21 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS22 OFF;
ALTER TABLE ORDERS22 DROP CONSTRAINT
ORDERS22CKC;
ALTER TABLE ORDERS22 ADD CONSTRAINT
ORDERS22CKC CHECK (O_W_ID BETWEEN 33601 AND
35200);
SET INTEGRITY FOR ORDERS22 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS23 OFF;
ALTER TABLE ORDERS23 DROP CONSTRAINT
ORDERS23CKC;
ALTER TABLE ORDERS23 ADD CONSTRAINT
ORDERS23CKC CHECK (O_W_ID BETWEEN 35201 AND
36800);
SET INTEGRITY FOR ORDERS23 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS24 OFF;
ALTER TABLE ORDERS24 DROP CONSTRAINT
ORDERS24CKC;
ALTER TABLE ORDERS24 ADD CONSTRAINT
ORDERS24CKC CHECK (O_W_ID BETWEEN 36801 AND
38400);
SET INTEGRITY FOR ORDERS24 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS25 OFF;
ALTER TABLE ORDERS25 DROP CONSTRAINT
ORDERS25CKC;
ALTER TABLE ORDERS25 ADD CONSTRAINT
ORDERS25CKC CHECK (O_W_ID BETWEEN 38401 AND
40000);
SET INTEGRITY FOR ORDERS25 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS26 OFF;
ALTER TABLE ORDERS26 DROP CONSTRAINT
ORDERS26CKC;

```

```

ALTER TABLE ORDERS26 ADD CONSTRAINT
ORDERS26CKC CHECK (O_W_ID BETWEEN 40001 AND
41600);
SET INTEGRITY FOR ORDERS26 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS27 OFF;
ALTER TABLE ORDERS27 DROP CONSTRAINT
ORDERS27CKC;
ALTER TABLE ORDERS27 ADD CONSTRAINT
ORDERS27CKC CHECK (O_W_ID BETWEEN 41601 AND
43200);
SET INTEGRITY FOR ORDERS27 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS28 OFF;
ALTER TABLE ORDERS28 DROP CONSTRAINT
ORDERS28CKC;
ALTER TABLE ORDERS28 ADD CONSTRAINT
ORDERS28CKC CHECK (O_W_ID BETWEEN 43201 AND
44800);
SET INTEGRITY FOR ORDERS28 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS29 OFF;
ALTER TABLE ORDERS29 DROP CONSTRAINT
ORDERS29CKC;
ALTER TABLE ORDERS29 ADD CONSTRAINT
ORDERS29CKC CHECK (O_W_ID BETWEEN 44801 AND
46400);
SET INTEGRITY FOR ORDERS29 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS30 OFF;
ALTER TABLE ORDERS30 DROP CONSTRAINT
ORDERS30CKC;
ALTER TABLE ORDERS30 ADD CONSTRAINT
ORDERS30CKC CHECK (O_W_ID BETWEEN 46401 AND
48000);
SET INTEGRITY FOR ORDERS30 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS31 OFF;
ALTER TABLE ORDERS31 DROP CONSTRAINT
ORDERS31CKC;
ALTER TABLE ORDERS31 ADD CONSTRAINT
ORDERS31CKC CHECK (O_W_ID BETWEEN 48001 AND
49600);
SET INTEGRITY FOR ORDERS31 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS32 OFF;
ALTER TABLE ORDERS32 DROP CONSTRAINT
ORDERS32CKC;

```

```

ALTER TABLE ORDERS32 ADD CONSTRAINT
ORDERS32CKC CHECK (O_W_ID BETWEEN 49601 AND
51200);
SET INTEGRITY FOR ORDERS32 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS33 OFF;
ALTER TABLE ORDERS33 DROP CONSTRAINT
ORDERS33CKC;
ALTER TABLE ORDERS33 ADD CONSTRAINT
ORDERS33CKC CHECK (O_W_ID BETWEEN 51201 AND
52800);
SET INTEGRITY FOR ORDERS33 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS34 OFF;
ALTER TABLE ORDERS34 DROP CONSTRAINT
ORDERS34CKC;
ALTER TABLE ORDERS34 ADD CONSTRAINT
ORDERS34CKC CHECK (O_W_ID BETWEEN 52801 AND
54400);
SET INTEGRITY FOR ORDERS34 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS35 OFF;
ALTER TABLE ORDERS35 DROP CONSTRAINT
ORDERS35CKC;
ALTER TABLE ORDERS35 ADD CONSTRAINT
ORDERS35CKC CHECK (O_W_ID BETWEEN 54401 AND
56000);
SET INTEGRITY FOR ORDERS35 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS36 OFF;
ALTER TABLE ORDERS36 DROP CONSTRAINT
ORDERS36CKC;
ALTER TABLE ORDERS36 ADD CONSTRAINT
ORDERS36CKC CHECK (O_W_ID BETWEEN 56001 AND
57600);
SET INTEGRITY FOR ORDERS36 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS37 OFF;
ALTER TABLE ORDERS37 DROP CONSTRAINT
ORDERS37CKC;
ALTER TABLE ORDERS37 ADD CONSTRAINT
ORDERS37CKC CHECK (O_W_ID BETWEEN 57601 AND
59200);
SET INTEGRITY FOR ORDERS37 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS38 OFF;
ALTER TABLE ORDERS38 DROP CONSTRAINT
ORDERS38CKC;

```

```

ALTER TABLE ORDERS38 ADD CONSTRAINT
ORDERS38CKC CHECK (O_W_ID BETWEEN 59201 AND
60800);
SET INTEGRITY FOR ORDERS38 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS39 OFF;
ALTER TABLE ORDERS39 DROP CONSTRAINT
ORDERS39CKC;
ALTER TABLE ORDERS39 ADD CONSTRAINT
ORDERS39CKC CHECK (O_W_ID BETWEEN 60801 AND
62400);
SET INTEGRITY FOR ORDERS39 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS40 OFF;
ALTER TABLE ORDERS40 DROP CONSTRAINT
ORDERS40CKC;
ALTER TABLE ORDERS40 ADD CONSTRAINT
ORDERS40CKC CHECK (O_W_ID BETWEEN 62401 AND
64000);
SET INTEGRITY FOR ORDERS40 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS41 OFF;
ALTER TABLE ORDERS41 DROP CONSTRAINT
ORDERS41CKC;
ALTER TABLE ORDERS41 ADD CONSTRAINT
ORDERS41CKC CHECK (O_W_ID BETWEEN 64001 AND
65600);
SET INTEGRITY FOR ORDERS41 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS42 OFF;
ALTER TABLE ORDERS42 DROP CONSTRAINT
ORDERS42CKC;
ALTER TABLE ORDERS42 ADD CONSTRAINT
ORDERS42CKC CHECK (O_W_ID BETWEEN 65601 AND
67200);
SET INTEGRITY FOR ORDERS42 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS43 OFF;
ALTER TABLE ORDERS43 DROP CONSTRAINT
ORDERS43CKC;
ALTER TABLE ORDERS43 ADD CONSTRAINT
ORDERS43CKC CHECK (O_W_ID BETWEEN 67201 AND
68800);
SET INTEGRITY FOR ORDERS43 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS44 OFF;
ALTER TABLE ORDERS44 DROP CONSTRAINT
ORDERS44CKC;

```

```

ALTER TABLE ORDERS44 ADD CONSTRAINT
ORDERS44CKC CHECK (O_W_ID BETWEEN 68801 AND
70400);
SET INTEGRITY FOR ORDERS44 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS45 OFF;
ALTER TABLE ORDERS45 DROP CONSTRAINT
ORDERS45CKC;
ALTER TABLE ORDERS45 ADD CONSTRAINT
ORDERS45CKC CHECK (O_W_ID BETWEEN 70401 AND
72000);
SET INTEGRITY FOR ORDERS45 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS46 OFF;
ALTER TABLE ORDERS46 DROP CONSTRAINT
ORDERS46CKC;
ALTER TABLE ORDERS46 ADD CONSTRAINT
ORDERS46CKC CHECK (O_W_ID BETWEEN 72001 AND
73600);
SET INTEGRITY FOR ORDERS46 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS47 OFF;
ALTER TABLE ORDERS47 DROP CONSTRAINT
ORDERS47CKC;
ALTER TABLE ORDERS47 ADD CONSTRAINT
ORDERS47CKC CHECK (O_W_ID BETWEEN 73601 AND
75200);
SET INTEGRITY FOR ORDERS47 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS48 OFF;
ALTER TABLE ORDERS48 DROP CONSTRAINT
ORDERS48CKC;
ALTER TABLE ORDERS48 ADD CONSTRAINT
ORDERS48CKC CHECK (O_W_ID BETWEEN 75201 AND
76800);
SET INTEGRITY FOR ORDERS48 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS49 OFF;
ALTER TABLE ORDERS49 DROP CONSTRAINT
ORDERS49CKC;
ALTER TABLE ORDERS49 ADD CONSTRAINT
ORDERS49CKC CHECK (O_W_ID BETWEEN 76801 AND
78400);
SET INTEGRITY FOR ORDERS49 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS50 OFF;
ALTER TABLE ORDERS50 DROP CONSTRAINT
ORDERS50CKC;

```



```

ALTER TABLE ORDERS50 ADD CONSTRAINT
ORDERS50CKC CHECK (O_W_ID BETWEEN 78401 AND
80000);
SET INTEGRITY FOR ORDERS50 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS51 OFF;
ALTER TABLE ORDERS51 DROP CONSTRAINT
ORDERS51CKC;
ALTER TABLE ORDERS51 ADD CONSTRAINT
ORDERS51CKC CHECK (O_W_ID >= 80001);
SET INTEGRITY FOR ORDERS51 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

DDL/CRCONST ORDER LINE.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE1 OFF;
ALTER TABLE ORDER_LINE1 DROP CONSTRAINT
ORDER_LINE1CKC;
ALTER TABLE ORDER_LINE1 ADD CONSTRAINT
ORDER_LINE1CKC CHECK (OL_W_ID BETWEEN 1 AND
1600);
SET INTEGRITY FOR ORDER_LINE1 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE2 OFF;
ALTER TABLE ORDER_LINE2 DROP CONSTRAINT
ORDER_LINE2CKC;
ALTER TABLE ORDER_LINE2 ADD CONSTRAINT
ORDER_LINE2CKC CHECK (OL_W_ID BETWEEN 1601 AND
3200);
SET INTEGRITY FOR ORDER_LINE2 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE3 OFF;
ALTER TABLE ORDER_LINE3 DROP CONSTRAINT
ORDER_LINE3CKC;
ALTER TABLE ORDER_LINE3 ADD CONSTRAINT
ORDER_LINE3CKC CHECK (OL_W_ID BETWEEN 3201 AND
4800);
SET INTEGRITY FOR ORDER_LINE3 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE4 OFF;
ALTER TABLE ORDER_LINE4 DROP CONSTRAINT
ORDER_LINE4CKC;
ALTER TABLE ORDER_LINE4 ADD CONSTRAINT
ORDER_LINE4CKC CHECK (OL_W_ID BETWEEN 4801 AND
6400);
SET INTEGRITY FOR ORDER_LINE4 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE5 OFF;

```

```

ALTER TABLE ORDER_LINE5 DROP CONSTRAINT
ORDER_LINE5CKC;
ALTER TABLE ORDER_LINE5 ADD CONSTRAINT
ORDER_LINE5CKC CHECK (OL_W_ID BETWEEN 6401 AND
8000);
SET INTEGRITY FOR ORDER_LINE5 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE6 OFF;
ALTER TABLE ORDER_LINE6 DROP CONSTRAINT
ORDER_LINE6CKC;
ALTER TABLE ORDER_LINE6 ADD CONSTRAINT
ORDER_LINE6CKC CHECK (OL_W_ID BETWEEN 8001 AND
9600);
SET INTEGRITY FOR ORDER_LINE6 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE7 OFF;
ALTER TABLE ORDER_LINE7 DROP CONSTRAINT
ORDER_LINE7CKC;
ALTER TABLE ORDER_LINE7 ADD CONSTRAINT
ORDER_LINE7CKC CHECK (OL_W_ID BETWEEN 9601 AND
11200);
SET INTEGRITY FOR ORDER_LINE7 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE8 OFF;
ALTER TABLE ORDER_LINE8 DROP CONSTRAINT
ORDER_LINE8CKC;
ALTER TABLE ORDER_LINE8 ADD CONSTRAINT
ORDER_LINE8CKC CHECK (OL_W_ID BETWEEN 11201
AND 12800);
SET INTEGRITY FOR ORDER_LINE8 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE9 OFF;
ALTER TABLE ORDER_LINE9 DROP CONSTRAINT
ORDER_LINE9CKC;
ALTER TABLE ORDER_LINE9 ADD CONSTRAINT
ORDER_LINE9CKC CHECK (OL_W_ID BETWEEN 12801
AND 14400);
SET INTEGRITY FOR ORDER_LINE9 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE10 OFF;
ALTER TABLE ORDER_LINE10 DROP CONSTRAINT
ORDER_LINE10CKC;
ALTER TABLE ORDER_LINE10 ADD CONSTRAINT
ORDER_LINE10CKC CHECK (OL_W_ID BETWEEN 14401
AND 16000);
SET INTEGRITY FOR ORDER_LINE10 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE11 OFF;
ALTER TABLE ORDER_LINE11 DROP CONSTRAINT
ORDER_LINE11CKC;

```

```

ALTER TABLE ORDER_LINE11 ADD CONSTRAINT
ORDER_LINE11CKC CHECK (OL_W_ID BETWEEN 16001
AND 17600);
SET INTEGRITY FOR ORDER_LINE11 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE12 OFF;
ALTER TABLE ORDER_LINE12 DROP CONSTRAINT
ORDER_LINE12CKC;
ALTER TABLE ORDER_LINE12 ADD CONSTRAINT
ORDER_LINE12CKC CHECK (OL_W_ID BETWEEN 17601
AND 19200);
SET INTEGRITY FOR ORDER_LINE12 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE13 OFF;
ALTER TABLE ORDER_LINE13 DROP CONSTRAINT
ORDER_LINE13CKC;
ALTER TABLE ORDER_LINE13 ADD CONSTRAINT
ORDER_LINE13CKC CHECK (OL_W_ID BETWEEN 19201
AND 20800);
SET INTEGRITY FOR ORDER_LINE13 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE14 OFF;
ALTER TABLE ORDER_LINE14 DROP CONSTRAINT
ORDER_LINE14CKC;
ALTER TABLE ORDER_LINE14 ADD CONSTRAINT
ORDER_LINE14CKC CHECK (OL_W_ID BETWEEN 20801
AND 22400);
SET INTEGRITY FOR ORDER_LINE14 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE15 OFF;
ALTER TABLE ORDER_LINE15 DROP CONSTRAINT
ORDER_LINE15CKC;
ALTER TABLE ORDER_LINE15 ADD CONSTRAINT
ORDER_LINE15CKC CHECK (OL_W_ID BETWEEN 22401
AND 24000);
SET INTEGRITY FOR ORDER_LINE15 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE16 OFF;
ALTER TABLE ORDER_LINE16 DROP CONSTRAINT
ORDER_LINE16CKC;
ALTER TABLE ORDER_LINE16 ADD CONSTRAINT
ORDER_LINE16CKC CHECK (OL_W_ID BETWEEN 24001
AND 25600);
SET INTEGRITY FOR ORDER_LINE16 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE17 OFF;
ALTER TABLE ORDER_LINE17 DROP CONSTRAINT
ORDER_LINE17CKC;

```

```

ALTER TABLE ORDER_LINE17 ADD CONSTRAINT
ORDER_LINE17CKC CHECK (OL_W_ID BETWEEN 25601
AND 27200);
SET INTEGRITY FOR ORDER_LINE17 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE18 OFF;
ALTER TABLE ORDER_LINE18 DROP CONSTRAINT
ORDER_LINE18CKC;
ALTER TABLE ORDER_LINE18 ADD CONSTRAINT
ORDER_LINE18CKC CHECK (OL_W_ID BETWEEN 27201
AND 28800);
SET INTEGRITY FOR ORDER_LINE18 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE19 OFF;
ALTER TABLE ORDER_LINE19 DROP CONSTRAINT
ORDER_LINE19CKC;
ALTER TABLE ORDER_LINE19 ADD CONSTRAINT
ORDER_LINE19CKC CHECK (OL_W_ID BETWEEN 28801
AND 30400);
SET INTEGRITY FOR ORDER_LINE19 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE20 OFF;
ALTER TABLE ORDER_LINE20 DROP CONSTRAINT
ORDER_LINE20CKC;
ALTER TABLE ORDER_LINE20 ADD CONSTRAINT
ORDER_LINE20CKC CHECK (OL_W_ID BETWEEN 30401
AND 32000);
SET INTEGRITY FOR ORDER_LINE20 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE21 OFF;
ALTER TABLE ORDER_LINE21 DROP CONSTRAINT
ORDER_LINE21CKC;
ALTER TABLE ORDER_LINE21 ADD CONSTRAINT
ORDER_LINE21CKC CHECK (OL_W_ID BETWEEN 32001
AND 33600);
SET INTEGRITY FOR ORDER_LINE21 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE22 OFF;
ALTER TABLE ORDER_LINE22 DROP CONSTRAINT
ORDER_LINE22CKC;
ALTER TABLE ORDER_LINE22 ADD CONSTRAINT
ORDER_LINE22CKC CHECK (OL_W_ID BETWEEN 33601
AND 35200);
SET INTEGRITY FOR ORDER_LINE22 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE23 OFF;
ALTER TABLE ORDER_LINE23 DROP CONSTRAINT
ORDER_LINE23CKC;

```

```

ALTER TABLE ORDER_LINE23 ADD CONSTRAINT
ORDER_LINE23CKC CHECK (OL_W_ID BETWEEN 35201
AND 36800);
SET INTEGRITY FOR ORDER_LINE23 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE24 OFF;
ALTER TABLE ORDER_LINE24 DROP CONSTRAINT
ORDER_LINE24CKC;
ALTER TABLE ORDER_LINE24 ADD CONSTRAINT
ORDER_LINE24CKC CHECK (OL_W_ID BETWEEN 36801
AND 38400);
SET INTEGRITY FOR ORDER_LINE24 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE25 OFF;
ALTER TABLE ORDER_LINE25 DROP CONSTRAINT
ORDER_LINE25CKC;
ALTER TABLE ORDER_LINE25 ADD CONSTRAINT
ORDER_LINE25CKC CHECK (OL_W_ID BETWEEN 38401
AND 40000);
SET INTEGRITY FOR ORDER_LINE25 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE26 OFF;
ALTER TABLE ORDER_LINE26 DROP CONSTRAINT
ORDER_LINE26CKC;
ALTER TABLE ORDER_LINE26 ADD CONSTRAINT
ORDER_LINE26CKC CHECK (OL_W_ID BETWEEN 40001
AND 41600);
SET INTEGRITY FOR ORDER_LINE26 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE27 OFF;
ALTER TABLE ORDER_LINE27 DROP CONSTRAINT
ORDER_LINE27CKC;
ALTER TABLE ORDER_LINE27 ADD CONSTRAINT
ORDER_LINE27CKC CHECK (OL_W_ID BETWEEN 41601
AND 43200);
SET INTEGRITY FOR ORDER_LINE27 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE28 OFF;
ALTER TABLE ORDER_LINE28 DROP CONSTRAINT
ORDER_LINE28CKC;
ALTER TABLE ORDER_LINE28 ADD CONSTRAINT
ORDER_LINE28CKC CHECK (OL_W_ID BETWEEN 43201
AND 44800);
SET INTEGRITY FOR ORDER_LINE28 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE29 OFF;
ALTER TABLE ORDER_LINE29 DROP CONSTRAINT
ORDER_LINE29CKC;

```

```

ALTER TABLE ORDER_LINE29 ADD CONSTRAINT
ORDER_LINE29CKC CHECK (OL_W_ID BETWEEN 44801
AND 46400);
SET INTEGRITY FOR ORDER_LINE29 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE30 OFF;
ALTER TABLE ORDER_LINE30 DROP CONSTRAINT
ORDER_LINE30CKC;
ALTER TABLE ORDER_LINE30 ADD CONSTRAINT
ORDER_LINE30CKC CHECK (OL_W_ID BETWEEN 46401
AND 48000);
SET INTEGRITY FOR ORDER_LINE30 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE31 OFF;
ALTER TABLE ORDER_LINE31 DROP CONSTRAINT
ORDER_LINE31CKC;
ALTER TABLE ORDER_LINE31 ADD CONSTRAINT
ORDER_LINE31CKC CHECK (OL_W_ID BETWEEN 48001
AND 49600);
SET INTEGRITY FOR ORDER_LINE31 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE32 OFF;
ALTER TABLE ORDER_LINE32 DROP CONSTRAINT
ORDER_LINE32CKC;
ALTER TABLE ORDER_LINE32 ADD CONSTRAINT
ORDER_LINE32CKC CHECK (OL_W_ID BETWEEN 49601
AND 51200);
SET INTEGRITY FOR ORDER_LINE32 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE33 OFF;
ALTER TABLE ORDER_LINE33 DROP CONSTRAINT
ORDER_LINE33CKC;
ALTER TABLE ORDER_LINE33 ADD CONSTRAINT
ORDER_LINE33CKC CHECK (OL_W_ID BETWEEN 51201
AND 52800);
SET INTEGRITY FOR ORDER_LINE33 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE34 OFF;
ALTER TABLE ORDER_LINE34 DROP CONSTRAINT
ORDER_LINE34CKC;
ALTER TABLE ORDER_LINE34 ADD CONSTRAINT
ORDER_LINE34CKC CHECK (OL_W_ID BETWEEN 52801
AND 54400);
SET INTEGRITY FOR ORDER_LINE34 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE35 OFF;
ALTER TABLE ORDER_LINE35 DROP CONSTRAINT
ORDER_LINE35CKC;

```

```

ALTER TABLE ORDER_LINE35 ADD CONSTRAINT
ORDER_LINE35CKC CHECK (OL_W_ID BETWEEN 54401
AND 56000);
SET INTEGRITY FOR ORDER_LINE35 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE36 OFF;
ALTER TABLE ORDER_LINE36 DROP CONSTRAINT
ORDER_LINE36CKC;
ALTER TABLE ORDER_LINE36 ADD CONSTRAINT
ORDER_LINE36CKC CHECK (OL_W_ID BETWEEN 56001
AND 57600);
SET INTEGRITY FOR ORDER_LINE36 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE37 OFF;
ALTER TABLE ORDER_LINE37 DROP CONSTRAINT
ORDER_LINE37CKC;
ALTER TABLE ORDER_LINE37 ADD CONSTRAINT
ORDER_LINE37CKC CHECK (OL_W_ID BETWEEN 57601
AND 59200);
SET INTEGRITY FOR ORDER_LINE37 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE38 OFF;
ALTER TABLE ORDER_LINE38 DROP CONSTRAINT
ORDER_LINE38CKC;
ALTER TABLE ORDER_LINE38 ADD CONSTRAINT
ORDER_LINE38CKC CHECK (OL_W_ID BETWEEN 59201
AND 60800);
SET INTEGRITY FOR ORDER_LINE38 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE39 OFF;
ALTER TABLE ORDER_LINE39 DROP CONSTRAINT
ORDER_LINE39CKC;
ALTER TABLE ORDER_LINE39 ADD CONSTRAINT
ORDER_LINE39CKC CHECK (OL_W_ID BETWEEN 60801
AND 62400);
SET INTEGRITY FOR ORDER_LINE39 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE40 OFF;
ALTER TABLE ORDER_LINE40 DROP CONSTRAINT
ORDER_LINE40CKC;
ALTER TABLE ORDER_LINE40 ADD CONSTRAINT
ORDER_LINE40CKC CHECK (OL_W_ID BETWEEN 62401
AND 64000);
SET INTEGRITY FOR ORDER_LINE40 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE41 OFF;
ALTER TABLE ORDER_LINE41 DROP CONSTRAINT
ORDER_LINE41CKC;

```

```

ALTER TABLE ORDER_LINE41 ADD CONSTRAINT
ORDER_LINE41CKC CHECK (OL_W_ID BETWEEN 64001
AND 65600);
SET INTEGRITY FOR ORDER_LINE41 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE42 OFF;
ALTER TABLE ORDER_LINE42 DROP CONSTRAINT
ORDER_LINE42CKC;
ALTER TABLE ORDER_LINE42 ADD CONSTRAINT
ORDER_LINE42CKC CHECK (OL_W_ID BETWEEN 65601
AND 67200);
SET INTEGRITY FOR ORDER_LINE42 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE43 OFF;
ALTER TABLE ORDER_LINE43 DROP CONSTRAINT
ORDER_LINE43CKC;
ALTER TABLE ORDER_LINE43 ADD CONSTRAINT
ORDER_LINE43CKC CHECK (OL_W_ID BETWEEN 67201
AND 68800);
SET INTEGRITY FOR ORDER_LINE43 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE44 OFF;
ALTER TABLE ORDER_LINE44 DROP CONSTRAINT
ORDER_LINE44CKC;
ALTER TABLE ORDER_LINE44 ADD CONSTRAINT
ORDER_LINE44CKC CHECK (OL_W_ID BETWEEN 68801
AND 70400);
SET INTEGRITY FOR ORDER_LINE44 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE45 OFF;
ALTER TABLE ORDER_LINE45 DROP CONSTRAINT
ORDER_LINE45CKC;
ALTER TABLE ORDER_LINE45 ADD CONSTRAINT
ORDER_LINE45CKC CHECK (OL_W_ID BETWEEN 70401
AND 72000);
SET INTEGRITY FOR ORDER_LINE45 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE46 OFF;
ALTER TABLE ORDER_LINE46 DROP CONSTRAINT
ORDER_LINE46CKC;
ALTER TABLE ORDER_LINE46 ADD CONSTRAINT
ORDER_LINE46CKC CHECK (OL_W_ID BETWEEN 72001
AND 73600);
SET INTEGRITY FOR ORDER_LINE46 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE47 OFF;
ALTER TABLE ORDER_LINE47 DROP CONSTRAINT
ORDER_LINE47CKC;

```

```

ALTER TABLE ORDER_LINE47 ADD CONSTRAINT
ORDER_LINE47CKC CHECK (OL_W_ID BETWEEN 73601
AND 75200);
SET INTEGRITY FOR ORDER_LINE47 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE48 OFF;
ALTER TABLE ORDER_LINE48 DROP CONSTRAINT
ORDER_LINE48CKC;
ALTER TABLE ORDER_LINE48 ADD CONSTRAINT
ORDER_LINE48CKC CHECK (OL_W_ID BETWEEN 75201
AND 76800);
SET INTEGRITY FOR ORDER_LINE48 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE49 OFF;
ALTER TABLE ORDER_LINE49 DROP CONSTRAINT
ORDER_LINE49CKC;
ALTER TABLE ORDER_LINE49 ADD CONSTRAINT
ORDER_LINE49CKC CHECK (OL_W_ID BETWEEN 76801
AND 78400);
SET INTEGRITY FOR ORDER_LINE49 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE50 OFF;
ALTER TABLE ORDER_LINE50 DROP CONSTRAINT
ORDER_LINE50CKC;
ALTER TABLE ORDER_LINE50 ADD CONSTRAINT
ORDER_LINE50CKC CHECK (OL_W_ID BETWEEN 78401
AND 80000);
SET INTEGRITY FOR ORDER_LINE50 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE51 OFF;
ALTER TABLE ORDER_LINE51 DROP CONSTRAINT
ORDER_LINE51CKC;
ALTER TABLE ORDER_LINE51 ADD CONSTRAINT
ORDER_LINE51CKC CHECK (OL_W_ID >= 80001);
SET INTEGRITY FOR ORDER_LINE51 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

DDL/CRCONST STOCK.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK1 OFF;
ALTER TABLE STOCK1 DROP CONSTRAINT STOCK1CKC;
ALTER TABLE STOCK1 ADD CONSTRAINT STOCK1CKC
CHECK (S_W_ID BETWEEN 1 AND 1600);
SET INTEGRITY FOR STOCK1 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK2 OFF;
ALTER TABLE STOCK2 DROP CONSTRAINT STOCK2CKC;
ALTER TABLE STOCK2 ADD CONSTRAINT STOCK2CKC
CHECK (S_W_ID BETWEEN 1601 AND 3200);

```

```

SET INTEGRITY FOR STOCK2 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK3 OFF;
ALTER TABLE STOCK3 DROP CONSTRAINT STOCK3CKC;
ALTER TABLE STOCK3 ADD CONSTRAINT STOCK3CKC
CHECK (S_W_ID BETWEEN 3201 AND 4800);
SET INTEGRITY FOR STOCK3 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK4 OFF;
ALTER TABLE STOCK4 DROP CONSTRAINT STOCK4CKC;
ALTER TABLE STOCK4 ADD CONSTRAINT STOCK4CKC
CHECK (S_W_ID BETWEEN 4801 AND 6400);
SET INTEGRITY FOR STOCK4 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK5 OFF;
ALTER TABLE STOCK5 DROP CONSTRAINT STOCK5CKC;
ALTER TABLE STOCK5 ADD CONSTRAINT STOCK5CKC
CHECK (S_W_ID BETWEEN 6401 AND 8000);
SET INTEGRITY FOR STOCK5 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK6 OFF;
ALTER TABLE STOCK6 DROP CONSTRAINT STOCK6CKC;
ALTER TABLE STOCK6 ADD CONSTRAINT STOCK6CKC
CHECK (S_W_ID BETWEEN 8001 AND 9600);
SET INTEGRITY FOR STOCK6 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK7 OFF;
ALTER TABLE STOCK7 DROP CONSTRAINT STOCK7CKC;
ALTER TABLE STOCK7 ADD CONSTRAINT STOCK7CKC
CHECK (S_W_ID BETWEEN 9601 AND 11200);
SET INTEGRITY FOR STOCK7 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK8 OFF;
ALTER TABLE STOCK8 DROP CONSTRAINT STOCK8CKC;
ALTER TABLE STOCK8 ADD CONSTRAINT STOCK8CKC
CHECK (S_W_ID BETWEEN 11201 AND 12800);
SET INTEGRITY FOR STOCK8 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK9 OFF;
ALTER TABLE STOCK9 DROP CONSTRAINT STOCK9CKC;
ALTER TABLE STOCK9 ADD CONSTRAINT STOCK9CKC
CHECK (S_W_ID BETWEEN 12801 AND 14400);
SET INTEGRITY FOR STOCK9 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK10 OFF;

```

```

ALTER TABLE STOCK10 DROP CONSTRAINT
STOCK10CKC;
ALTER TABLE STOCK10 ADD CONSTRAINT STOCK10CKC
CHECK (S_W_ID BETWEEN 14401 AND 16000);
SET INTEGRITY FOR STOCK10 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK11 OFF;
ALTER TABLE STOCK11 DROP CONSTRAINT
STOCK11CKC;
ALTER TABLE STOCK11 ADD CONSTRAINT STOCK11CKC
CHECK (S_W_ID BETWEEN 16001 AND 17600);
SET INTEGRITY FOR STOCK11 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK12 OFF;
ALTER TABLE STOCK12 DROP CONSTRAINT
STOCK12CKC;
ALTER TABLE STOCK12 ADD CONSTRAINT STOCK12CKC
CHECK (S_W_ID BETWEEN 17601 AND 19200);
SET INTEGRITY FOR STOCK12 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK13 OFF;
ALTER TABLE STOCK13 DROP CONSTRAINT
STOCK13CKC;
ALTER TABLE STOCK13 ADD CONSTRAINT STOCK13CKC
CHECK (S_W_ID BETWEEN 19201 AND 20800);
SET INTEGRITY FOR STOCK13 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK14 OFF;
ALTER TABLE STOCK14 DROP CONSTRAINT
STOCK14CKC;
ALTER TABLE STOCK14 ADD CONSTRAINT STOCK14CKC
CHECK (S_W_ID BETWEEN 20801 AND 22400);
SET INTEGRITY FOR STOCK14 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK15 OFF;
ALTER TABLE STOCK15 DROP CONSTRAINT
STOCK15CKC;
ALTER TABLE STOCK15 ADD CONSTRAINT STOCK15CKC
CHECK (S_W_ID BETWEEN 22401 AND 24000);
SET INTEGRITY FOR STOCK15 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK16 OFF;
ALTER TABLE STOCK16 DROP CONSTRAINT
STOCK16CKC;
ALTER TABLE STOCK16 ADD CONSTRAINT STOCK16CKC
CHECK (S_W_ID BETWEEN 24001 AND 25600);
SET INTEGRITY FOR STOCK16 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;

```

```

SET INTEGRITY FOR STOCK17 OFF;
ALTER TABLE STOCK17 DROP CONSTRAINT
STOCK17CKC;
ALTER TABLE STOCK17 ADD CONSTRAINT STOCK17CKC
CHECK (S_W_ID BETWEEN 25601 AND 27200);
SET INTEGRITY FOR STOCK17 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK18 OFF;
ALTER TABLE STOCK18 DROP CONSTRAINT
STOCK18CKC;
ALTER TABLE STOCK18 ADD CONSTRAINT STOCK18CKC
CHECK (S_W_ID BETWEEN 27201 AND 28800);
SET INTEGRITY FOR STOCK18 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK19 OFF;
ALTER TABLE STOCK19 DROP CONSTRAINT
STOCK19CKC;
ALTER TABLE STOCK19 ADD CONSTRAINT STOCK19CKC
CHECK (S_W_ID BETWEEN 28801 AND 30400);
SET INTEGRITY FOR STOCK19 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK20 OFF;
ALTER TABLE STOCK20 DROP CONSTRAINT
STOCK20CKC;
ALTER TABLE STOCK20 ADD CONSTRAINT STOCK20CKC
CHECK (S_W_ID BETWEEN 30401 AND 32000);
SET INTEGRITY FOR STOCK20 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK21 OFF;
ALTER TABLE STOCK21 DROP CONSTRAINT
STOCK21CKC;
ALTER TABLE STOCK21 ADD CONSTRAINT STOCK21CKC
CHECK (S_W_ID BETWEEN 32001 AND 33600);
SET INTEGRITY FOR STOCK21 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK22 OFF;
ALTER TABLE STOCK22 DROP CONSTRAINT
STOCK22CKC;
ALTER TABLE STOCK22 ADD CONSTRAINT STOCK22CKC
CHECK (S_W_ID BETWEEN 33601 AND 35200);
SET INTEGRITY FOR STOCK22 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK23 OFF;
ALTER TABLE STOCK23 DROP CONSTRAINT
STOCK23CKC;
ALTER TABLE STOCK23 ADD CONSTRAINT STOCK23CKC
CHECK (S_W_ID BETWEEN 35201 AND 36800);
SET INTEGRITY FOR STOCK23 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

```

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK24 OFF;
ALTER TABLE STOCK24 DROP CONSTRAINT
STOCK24CKC;
ALTER TABLE STOCK24 ADD CONSTRAINT STOCK24CKC
CHECK (S_W_ID BETWEEN 36801 AND 38400);
SET INTEGRITY FOR STOCK24 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK25 OFF;
ALTER TABLE STOCK25 DROP CONSTRAINT
STOCK25CKC;
ALTER TABLE STOCK25 ADD CONSTRAINT STOCK25CKC
CHECK (S_W_ID BETWEEN 38401 AND 40000);
SET INTEGRITY FOR STOCK25 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK26 OFF;
ALTER TABLE STOCK26 DROP CONSTRAINT
STOCK26CKC;
ALTER TABLE STOCK26 ADD CONSTRAINT STOCK26CKC
CHECK (S_W_ID BETWEEN 40001 AND 41600);
SET INTEGRITY FOR STOCK26 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK27 OFF;
ALTER TABLE STOCK27 DROP CONSTRAINT
STOCK27CKC;
ALTER TABLE STOCK27 ADD CONSTRAINT STOCK27CKC
CHECK (S_W_ID BETWEEN 41601 AND 43200);
SET INTEGRITY FOR STOCK27 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK28 OFF;
ALTER TABLE STOCK28 DROP CONSTRAINT
STOCK28CKC;
ALTER TABLE STOCK28 ADD CONSTRAINT STOCK28CKC
CHECK (S_W_ID BETWEEN 43201 AND 44800);
SET INTEGRITY FOR STOCK28 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK29 OFF;
ALTER TABLE STOCK29 DROP CONSTRAINT
STOCK29CKC;
ALTER TABLE STOCK29 ADD CONSTRAINT STOCK29CKC
CHECK (S_W_ID BETWEEN 44801 AND 46400);
SET INTEGRITY FOR STOCK29 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK30 OFF;
ALTER TABLE STOCK30 DROP CONSTRAINT
STOCK30CKC;
ALTER TABLE STOCK30 ADD CONSTRAINT STOCK30CKC
CHECK (S_W_ID BETWEEN 46401 AND 48000);
SET INTEGRITY FOR STOCK30 ALL IMMEDIATE
UNCHECKED;

```

```

connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK31 OFF;
ALTER TABLE STOCK31 DROP CONSTRAINT
STOCK31CKC;
ALTER TABLE STOCK31 ADD CONSTRAINT STOCK31CKC
CHECK (S_W_ID BETWEEN 48001 AND 49600);
SET INTEGRITY FOR STOCK31 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK32 OFF;
ALTER TABLE STOCK32 DROP CONSTRAINT
STOCK32CKC;
ALTER TABLE STOCK32 ADD CONSTRAINT STOCK32CKC
CHECK (S_W_ID BETWEEN 49601 AND 51200);
SET INTEGRITY FOR STOCK32 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK33 OFF;
ALTER TABLE STOCK33 DROP CONSTRAINT
STOCK33CKC;
ALTER TABLE STOCK33 ADD CONSTRAINT STOCK33CKC
CHECK (S_W_ID BETWEEN 51201 AND 52800);
SET INTEGRITY FOR STOCK33 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK34 OFF;
ALTER TABLE STOCK34 DROP CONSTRAINT
STOCK34CKC;
ALTER TABLE STOCK34 ADD CONSTRAINT STOCK34CKC
CHECK (S_W_ID BETWEEN 52801 AND 54400);
SET INTEGRITY FOR STOCK34 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK35 OFF;
ALTER TABLE STOCK35 DROP CONSTRAINT
STOCK35CKC;
ALTER TABLE STOCK35 ADD CONSTRAINT STOCK35CKC
CHECK (S_W_ID BETWEEN 54401 AND 56000);
SET INTEGRITY FOR STOCK35 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK36 OFF;
ALTER TABLE STOCK36 DROP CONSTRAINT
STOCK36CKC;
ALTER TABLE STOCK36 ADD CONSTRAINT STOCK36CKC
CHECK (S_W_ID BETWEEN 56001 AND 57600);
SET INTEGRITY FOR STOCK36 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK37 OFF;
ALTER TABLE STOCK37 DROP CONSTRAINT
STOCK37CKC;
ALTER TABLE STOCK37 ADD CONSTRAINT STOCK37CKC
CHECK (S_W_ID BETWEEN 57601 AND 59200);

```

```

SET INTEGRITY FOR STOCK37 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK38 OFF;
ALTER TABLE STOCK38 DROP CONSTRAINT
STOCK38CKC;
ALTER TABLE STOCK38 ADD CONSTRAINT STOCK38CKC
CHECK (S_W_ID BETWEEN 59201 AND 60800);
SET INTEGRITY FOR STOCK38 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK39 OFF;
ALTER TABLE STOCK39 DROP CONSTRAINT
STOCK39CKC;
ALTER TABLE STOCK39 ADD CONSTRAINT STOCK39CKC
CHECK (S_W_ID BETWEEN 60801 AND 62400);
SET INTEGRITY FOR STOCK39 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK40 OFF;
ALTER TABLE STOCK40 DROP CONSTRAINT
STOCK40CKC;
ALTER TABLE STOCK40 ADD CONSTRAINT STOCK40CKC
CHECK (S_W_ID BETWEEN 62401 AND 64000);
SET INTEGRITY FOR STOCK40 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK41 OFF;
ALTER TABLE STOCK41 DROP CONSTRAINT
STOCK41CKC;
ALTER TABLE STOCK41 ADD CONSTRAINT STOCK41CKC
CHECK (S_W_ID BETWEEN 64001 AND 65600);
SET INTEGRITY FOR STOCK41 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK42 OFF;
ALTER TABLE STOCK42 DROP CONSTRAINT
STOCK42CKC;
ALTER TABLE STOCK42 ADD CONSTRAINT STOCK42CKC
CHECK (S_W_ID BETWEEN 65601 AND 67200);
SET INTEGRITY FOR STOCK42 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK43 OFF;
ALTER TABLE STOCK43 DROP CONSTRAINT
STOCK43CKC;
ALTER TABLE STOCK43 ADD CONSTRAINT STOCK43CKC
CHECK (S_W_ID BETWEEN 67201 AND 68800);
SET INTEGRITY FOR STOCK43 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK44 OFF;
ALTER TABLE STOCK44 DROP CONSTRAINT
STOCK44CKC;

```

```

ALTER TABLE STOCK44 ADD CONSTRAINT STOCK44CKC
CHECK (S_W_ID BETWEEN 68801 AND 70400);
SET INTEGRITY FOR STOCK44 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK45 OFF;
ALTER TABLE STOCK45 DROP CONSTRAINT
STOCK45CKC;
ALTER TABLE STOCK45 ADD CONSTRAINT STOCK45CKC
CHECK (S_W_ID BETWEEN 70401 AND 72000);
SET INTEGRITY FOR STOCK45 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK46 OFF;
ALTER TABLE STOCK46 DROP CONSTRAINT
STOCK46CKC;
ALTER TABLE STOCK46 ADD CONSTRAINT STOCK46CKC
CHECK (S_W_ID BETWEEN 72001 AND 73600);
SET INTEGRITY FOR STOCK46 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK47 OFF;
ALTER TABLE STOCK47 DROP CONSTRAINT
STOCK47CKC;
ALTER TABLE STOCK47 ADD CONSTRAINT STOCK47CKC
CHECK (S_W_ID BETWEEN 73601 AND 75200);
SET INTEGRITY FOR STOCK47 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK48 OFF;
ALTER TABLE STOCK48 DROP CONSTRAINT
STOCK48CKC;
ALTER TABLE STOCK48 ADD CONSTRAINT STOCK48CKC
CHECK (S_W_ID BETWEEN 75201 AND 76800);
SET INTEGRITY FOR STOCK48 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK49 OFF;
ALTER TABLE STOCK49 DROP CONSTRAINT
STOCK49CKC;
ALTER TABLE STOCK49 ADD CONSTRAINT STOCK49CKC
CHECK (S_W_ID BETWEEN 76801 AND 78400);
SET INTEGRITY FOR STOCK49 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK50 OFF;
ALTER TABLE STOCK50 DROP CONSTRAINT
STOCK50CKC;
ALTER TABLE STOCK50 ADD CONSTRAINT STOCK50CKC
CHECK (S_W_ID BETWEEN 78401 AND 80000);
SET INTEGRITY FOR STOCK50 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK51 OFF;

```

```

ALTER TABLE STOCK51 DROP CONSTRAINT
STOCK51CKC;
ALTER TABLE STOCK51 ADD CONSTRAINT STOCK51CKC
CHECK (S_W_ID >= 80001);
SET INTEGRITY FOR STOCK51 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

DDL/CRCONST WAREHOUSE.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE1 OFF;
ALTER TABLE WAREHOUSE1 DROP CONSTRAINT
WAREHOUSE1CKC;
ALTER TABLE WAREHOUSE1 ADD CONSTRAINT
WAREHOUSE1CKC CHECK (W_ID BETWEEN 1 AND 1600);
SET INTEGRITY FOR WAREHOUSE1 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE2 OFF;
ALTER TABLE WAREHOUSE2 DROP CONSTRAINT
WAREHOUSE2CKC;
ALTER TABLE WAREHOUSE2 ADD CONSTRAINT
WAREHOUSE2CKC CHECK (W_ID BETWEEN 1601 AND
3200);
SET INTEGRITY FOR WAREHOUSE2 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE3 OFF;
ALTER TABLE WAREHOUSE3 DROP CONSTRAINT
WAREHOUSE3CKC;
ALTER TABLE WAREHOUSE3 ADD CONSTRAINT
WAREHOUSE3CKC CHECK (W_ID BETWEEN 3201 AND
4800);
SET INTEGRITY FOR WAREHOUSE3 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE4 OFF;
ALTER TABLE WAREHOUSE4 DROP CONSTRAINT
WAREHOUSE4CKC;
ALTER TABLE WAREHOUSE4 ADD CONSTRAINT
WAREHOUSE4CKC CHECK (W_ID BETWEEN 4801 AND
6400);
SET INTEGRITY FOR WAREHOUSE4 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE5 OFF;
ALTER TABLE WAREHOUSE5 DROP CONSTRAINT
WAREHOUSE5CKC;
ALTER TABLE WAREHOUSE5 ADD CONSTRAINT
WAREHOUSE5CKC CHECK (W_ID BETWEEN 6401 AND
8000);
SET INTEGRITY FOR WAREHOUSE5 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE6 OFF;

```

```

ALTER TABLE WAREHOUSE6 DROP CONSTRAINT
WAREHOUSE6CKC;
ALTER TABLE WAREHOUSE6 ADD CONSTRAINT
WAREHOUSE6CKC CHECK (W_ID BETWEEN 8001 AND
9600);
SET INTEGRITY FOR WAREHOUSE6 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE7 OFF;
ALTER TABLE WAREHOUSE7 DROP CONSTRAINT
WAREHOUSE7CKC;
ALTER TABLE WAREHOUSE7 ADD CONSTRAINT
WAREHOUSE7CKC CHECK (W_ID BETWEEN 9601 AND
11200);
SET INTEGRITY FOR WAREHOUSE7 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE8 OFF;
ALTER TABLE WAREHOUSE8 DROP CONSTRAINT
WAREHOUSE8CKC;
ALTER TABLE WAREHOUSE8 ADD CONSTRAINT
WAREHOUSE8CKC CHECK (W_ID BETWEEN 11201 AND
12800);
SET INTEGRITY FOR WAREHOUSE8 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE9 OFF;
ALTER TABLE WAREHOUSE9 DROP CONSTRAINT
WAREHOUSE9CKC;
ALTER TABLE WAREHOUSE9 ADD CONSTRAINT
WAREHOUSE9CKC CHECK (W_ID BETWEEN 12801 AND
14400);
SET INTEGRITY FOR WAREHOUSE9 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE10 OFF;
ALTER TABLE WAREHOUSE10 DROP CONSTRAINT
WAREHOUSE10CKC;
ALTER TABLE WAREHOUSE10 ADD CONSTRAINT
WAREHOUSE10CKC CHECK (W_ID BETWEEN 14401 AND
16000);
SET INTEGRITY FOR WAREHOUSE10 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE11 OFF;
ALTER TABLE WAREHOUSE11 DROP CONSTRAINT
WAREHOUSE11CKC;
ALTER TABLE WAREHOUSE11 ADD CONSTRAINT
WAREHOUSE11CKC CHECK (W_ID BETWEEN 16001 AND
17600);
SET INTEGRITY FOR WAREHOUSE11 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE12 OFF;
ALTER TABLE WAREHOUSE12 DROP CONSTRAINT
WAREHOUSE12CKC;

```

```

ALTER TABLE WAREHOUSE12 ADD CONSTRAINT
WAREHOUSE12CKC CHECK (W_ID BETWEEN 17601 AND
19200);
SET INTEGRITY FOR WAREHOUSE12 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE13 OFF;
ALTER TABLE WAREHOUSE13 DROP CONSTRAINT
WAREHOUSE13CKC;
ALTER TABLE WAREHOUSE13 ADD CONSTRAINT
WAREHOUSE13CKC CHECK (W_ID BETWEEN 19201 AND
20800);
SET INTEGRITY FOR WAREHOUSE13 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE14 OFF;
ALTER TABLE WAREHOUSE14 DROP CONSTRAINT
WAREHOUSE14CKC;
ALTER TABLE WAREHOUSE14 ADD CONSTRAINT
WAREHOUSE14CKC CHECK (W_ID BETWEEN 20801 AND
22400);
SET INTEGRITY FOR WAREHOUSE14 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE15 OFF;
ALTER TABLE WAREHOUSE15 DROP CONSTRAINT
WAREHOUSE15CKC;
ALTER TABLE WAREHOUSE15 ADD CONSTRAINT
WAREHOUSE15CKC CHECK (W_ID BETWEEN 22401 AND
24000);
SET INTEGRITY FOR WAREHOUSE15 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE16 OFF;
ALTER TABLE WAREHOUSE16 DROP CONSTRAINT
WAREHOUSE16CKC;
ALTER TABLE WAREHOUSE16 ADD CONSTRAINT
WAREHOUSE16CKC CHECK (W_ID BETWEEN 24001 AND
25600);
SET INTEGRITY FOR WAREHOUSE16 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE17 OFF;
ALTER TABLE WAREHOUSE17 DROP CONSTRAINT
WAREHOUSE17CKC;
ALTER TABLE WAREHOUSE17 ADD CONSTRAINT
WAREHOUSE17CKC CHECK (W_ID BETWEEN 25601 AND
27200);
SET INTEGRITY FOR WAREHOUSE17 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE18 OFF;
ALTER TABLE WAREHOUSE18 DROP CONSTRAINT
WAREHOUSE18CKC;

```

```

ALTER TABLE WAREHOUSE18 ADD CONSTRAINT
WAREHOUSE18CKC CHECK (W_ID BETWEEN 27201 AND
28800);
SET INTEGRITY FOR WAREHOUSE18 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE19 OFF;
ALTER TABLE WAREHOUSE19 DROP CONSTRAINT
WAREHOUSE19CKC;
ALTER TABLE WAREHOUSE19 ADD CONSTRAINT
WAREHOUSE19CKC CHECK (W_ID BETWEEN 28801 AND
30400);
SET INTEGRITY FOR WAREHOUSE19 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE20 OFF;
ALTER TABLE WAREHOUSE20 DROP CONSTRAINT
WAREHOUSE20CKC;
ALTER TABLE WAREHOUSE20 ADD CONSTRAINT
WAREHOUSE20CKC CHECK (W_ID BETWEEN 30401 AND
32000);
SET INTEGRITY FOR WAREHOUSE20 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE21 OFF;
ALTER TABLE WAREHOUSE21 DROP CONSTRAINT
WAREHOUSE21CKC;
ALTER TABLE WAREHOUSE21 ADD CONSTRAINT
WAREHOUSE21CKC CHECK (W_ID BETWEEN 32001 AND
33600);
SET INTEGRITY FOR WAREHOUSE21 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE22 OFF;
ALTER TABLE WAREHOUSE22 DROP CONSTRAINT
WAREHOUSE22CKC;
ALTER TABLE WAREHOUSE22 ADD CONSTRAINT
WAREHOUSE22CKC CHECK (W_ID BETWEEN 33601 AND
35200);
SET INTEGRITY FOR WAREHOUSE22 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE23 OFF;
ALTER TABLE WAREHOUSE23 DROP CONSTRAINT
WAREHOUSE23CKC;
ALTER TABLE WAREHOUSE23 ADD CONSTRAINT
WAREHOUSE23CKC CHECK (W_ID BETWEEN 35201 AND
36800);
SET INTEGRITY FOR WAREHOUSE23 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE24 OFF;
ALTER TABLE WAREHOUSE24 DROP CONSTRAINT
WAREHOUSE24CKC;

```

```

ALTER TABLE WAREHOUSE24 ADD CONSTRAINT
WAREHOUSE24CKC CHECK (W_ID BETWEEN 36801 AND
38400);
SET INTEGRITY FOR WAREHOUSE24 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE25 OFF;
ALTER TABLE WAREHOUSE25 DROP CONSTRAINT
WAREHOUSE25CKC;
ALTER TABLE WAREHOUSE25 ADD CONSTRAINT
WAREHOUSE25CKC CHECK (W_ID BETWEEN 38401 AND
40000);
SET INTEGRITY FOR WAREHOUSE25 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE26 OFF;
ALTER TABLE WAREHOUSE26 DROP CONSTRAINT
WAREHOUSE26CKC;
ALTER TABLE WAREHOUSE26 ADD CONSTRAINT
WAREHOUSE26CKC CHECK (W_ID BETWEEN 40001 AND
41600);
SET INTEGRITY FOR WAREHOUSE26 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE27 OFF;
ALTER TABLE WAREHOUSE27 DROP CONSTRAINT
WAREHOUSE27CKC;
ALTER TABLE WAREHOUSE27 ADD CONSTRAINT
WAREHOUSE27CKC CHECK (W_ID BETWEEN 41601 AND
43200);
SET INTEGRITY FOR WAREHOUSE27 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE28 OFF;
ALTER TABLE WAREHOUSE28 DROP CONSTRAINT
WAREHOUSE28CKC;
ALTER TABLE WAREHOUSE28 ADD CONSTRAINT
WAREHOUSE28CKC CHECK (W_ID BETWEEN 43201 AND
44800);
SET INTEGRITY FOR WAREHOUSE28 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE29 OFF;
ALTER TABLE WAREHOUSE29 DROP CONSTRAINT
WAREHOUSE29CKC;
ALTER TABLE WAREHOUSE29 ADD CONSTRAINT
WAREHOUSE29CKC CHECK (W_ID BETWEEN 44801 AND
46400);
SET INTEGRITY FOR WAREHOUSE29 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE30 OFF;
ALTER TABLE WAREHOUSE30 DROP CONSTRAINT
WAREHOUSE30CKC;

```

```

ALTER TABLE WAREHOUSE30 ADD CONSTRAINT
WAREHOUSE30CKC CHECK (W_ID BETWEEN 46401 AND
48000);
SET INTEGRITY FOR WAREHOUSE30 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE31 OFF;
ALTER TABLE WAREHOUSE31 DROP CONSTRAINT
WAREHOUSE31CKC;
ALTER TABLE WAREHOUSE31 ADD CONSTRAINT
WAREHOUSE31CKC CHECK (W_ID BETWEEN 48001 AND
49600);
SET INTEGRITY FOR WAREHOUSE31 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE32 OFF;
ALTER TABLE WAREHOUSE32 DROP CONSTRAINT
WAREHOUSE32CKC;
ALTER TABLE WAREHOUSE32 ADD CONSTRAINT
WAREHOUSE32CKC CHECK (W_ID BETWEEN 49601 AND
51200);
SET INTEGRITY FOR WAREHOUSE32 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE33 OFF;
ALTER TABLE WAREHOUSE33 DROP CONSTRAINT
WAREHOUSE33CKC;
ALTER TABLE WAREHOUSE33 ADD CONSTRAINT
WAREHOUSE33CKC CHECK (W_ID BETWEEN 51201 AND
52800);
SET INTEGRITY FOR WAREHOUSE33 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE34 OFF;
ALTER TABLE WAREHOUSE34 DROP CONSTRAINT
WAREHOUSE34CKC;
ALTER TABLE WAREHOUSE34 ADD CONSTRAINT
WAREHOUSE34CKC CHECK (W_ID BETWEEN 52801 AND
54400);
SET INTEGRITY FOR WAREHOUSE34 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE35 OFF;
ALTER TABLE WAREHOUSE35 DROP CONSTRAINT
WAREHOUSE35CKC;
ALTER TABLE WAREHOUSE35 ADD CONSTRAINT
WAREHOUSE35CKC CHECK (W_ID BETWEEN 54401 AND
56000);
SET INTEGRITY FOR WAREHOUSE35 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE36 OFF;
ALTER TABLE WAREHOUSE36 DROP CONSTRAINT
WAREHOUSE36CKC;

```

```

ALTER TABLE WAREHOUSE36 ADD CONSTRAINT
WAREHOUSE36CKC CHECK (W_ID BETWEEN 56001 AND
57600);
SET INTEGRITY FOR WAREHOUSE36 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE37 OFF;
ALTER TABLE WAREHOUSE37 DROP CONSTRAINT
WAREHOUSE37CKC;
ALTER TABLE WAREHOUSE37 ADD CONSTRAINT
WAREHOUSE37CKC CHECK (W_ID BETWEEN 57601 AND
59200);
SET INTEGRITY FOR WAREHOUSE37 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE38 OFF;
ALTER TABLE WAREHOUSE38 DROP CONSTRAINT
WAREHOUSE38CKC;
ALTER TABLE WAREHOUSE38 ADD CONSTRAINT
WAREHOUSE38CKC CHECK (W_ID BETWEEN 59201 AND
60800);
SET INTEGRITY FOR WAREHOUSE38 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE39 OFF;
ALTER TABLE WAREHOUSE39 DROP CONSTRAINT
WAREHOUSE39CKC;
ALTER TABLE WAREHOUSE39 ADD CONSTRAINT
WAREHOUSE39CKC CHECK (W_ID BETWEEN 60801 AND
62400);
SET INTEGRITY FOR WAREHOUSE39 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE40 OFF;
ALTER TABLE WAREHOUSE40 DROP CONSTRAINT
WAREHOUSE40CKC;
ALTER TABLE WAREHOUSE40 ADD CONSTRAINT
WAREHOUSE40CKC CHECK (W_ID BETWEEN 62401 AND
64000);
SET INTEGRITY FOR WAREHOUSE40 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE41 OFF;
ALTER TABLE WAREHOUSE41 DROP CONSTRAINT
WAREHOUSE41CKC;
ALTER TABLE WAREHOUSE41 ADD CONSTRAINT
WAREHOUSE41CKC CHECK (W_ID BETWEEN 64001 AND
65600);
SET INTEGRITY FOR WAREHOUSE41 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE42 OFF;
ALTER TABLE WAREHOUSE42 DROP CONSTRAINT
WAREHOUSE42CKC;

```

```

ALTER TABLE WAREHOUSE42 ADD CONSTRAINT
WAREHOUSE42CKC CHECK (W_ID BETWEEN 65601 AND
67200);
SET INTEGRITY FOR WAREHOUSE42 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE43 OFF;
ALTER TABLE WAREHOUSE43 DROP CONSTRAINT
WAREHOUSE43CKC;
ALTER TABLE WAREHOUSE43 ADD CONSTRAINT
WAREHOUSE43CKC CHECK (W_ID BETWEEN 67201 AND
68800);
SET INTEGRITY FOR WAREHOUSE43 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE44 OFF;
ALTER TABLE WAREHOUSE44 DROP CONSTRAINT
WAREHOUSE44CKC;
ALTER TABLE WAREHOUSE44 ADD CONSTRAINT
WAREHOUSE44CKC CHECK (W_ID BETWEEN 68801 AND
70400);
SET INTEGRITY FOR WAREHOUSE44 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE45 OFF;
ALTER TABLE WAREHOUSE45 DROP CONSTRAINT
WAREHOUSE45CKC;
ALTER TABLE WAREHOUSE45 ADD CONSTRAINT
WAREHOUSE45CKC CHECK (W_ID BETWEEN 70401 AND
72000);
SET INTEGRITY FOR WAREHOUSE45 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE46 OFF;
ALTER TABLE WAREHOUSE46 DROP CONSTRAINT
WAREHOUSE46CKC;
ALTER TABLE WAREHOUSE46 ADD CONSTRAINT
WAREHOUSE46CKC CHECK (W_ID BETWEEN 72001 AND
73600);
SET INTEGRITY FOR WAREHOUSE46 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE47 OFF;
ALTER TABLE WAREHOUSE47 DROP CONSTRAINT
WAREHOUSE47CKC;
ALTER TABLE WAREHOUSE47 ADD CONSTRAINT
WAREHOUSE47CKC CHECK (W_ID BETWEEN 73601 AND
75200);
SET INTEGRITY FOR WAREHOUSE47 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE48 OFF;
ALTER TABLE WAREHOUSE48 DROP CONSTRAINT
WAREHOUSE48CKC;

```



```

ALTER TABLE WAREHOUSE48 ADD CONSTRAINT
WAREHOUSE48CKC CHECK (W_ID BETWEEN 75201 AND
76800);
SET INTEGRITY FOR WAREHOUSE48 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE49 OFF;
ALTER TABLE WAREHOUSE49 DROP CONSTRAINT
WAREHOUSE49CKC;
ALTER TABLE WAREHOUSE49 ADD CONSTRAINT
WAREHOUSE49CKC CHECK (W_ID BETWEEN 76801 AND
78400);
SET INTEGRITY FOR WAREHOUSE49 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE50 OFF;
ALTER TABLE WAREHOUSE50 DROP CONSTRAINT
WAREHOUSE50CKC;
ALTER TABLE WAREHOUSE50 ADD CONSTRAINT
WAREHOUSE50CKC CHECK (W_ID BETWEEN 78401 AND
80000);
SET INTEGRITY FOR WAREHOUSE50 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE51 OFF;
ALTER TABLE WAREHOUSE51 DROP CONSTRAINT
WAREHOUSE51CKC;
ALTER TABLE WAREHOUSE51 ADD CONSTRAINT
WAREHOUSE51CKC CHECK (W_ID >= 80001);
SET INTEGRITY FOR WAREHOUSE51 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

DDL/CRIDX_CUST_IDXB.ddl

```

connect to TPCC in share mode;
DROP INDEX CUST_IDXB1;
CREATE INDEX CUST_IDXB1
      ON CUSTOMER1(C_LAST, C_W_ID, C_D_ID,
C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB2;
CREATE INDEX CUST_IDXB2
      ON CUSTOMER2(C_LAST, C_W_ID, C_D_ID,
C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB3;
CREATE INDEX CUST_IDXB3
      ON CUSTOMER3(C_LAST, C_W_ID, C_D_ID,
C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB4;
CREATE INDEX CUST_IDXB4
      ON CUSTOMER4(C_LAST, C_W_ID, C_D_ID,
C_FIRST, C_ID) PCTFREE 0;

```

```

connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB5;
CREATE INDEX CUST_IDXB5
      ON CUSTOMER5(C_LAST, C_W_ID, C_D_ID,
C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB6;
CREATE INDEX CUST_IDXB6
      ON CUSTOMER6(C_LAST, C_W_ID, C_D_ID,
C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB7;
CREATE INDEX CUST_IDXB7
      ON CUSTOMER7(C_LAST, C_W_ID, C_D_ID,
C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB8;
CREATE INDEX CUST_IDXB8
      ON CUSTOMER8(C_LAST, C_W_ID, C_D_ID,
C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB9;
CREATE INDEX CUST_IDXB9
      ON CUSTOMER9(C_LAST, C_W_ID, C_D_ID,
C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB10;
CREATE INDEX CUST_IDXB10
      ON CUSTOMER10(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB11;
CREATE INDEX CUST_IDXB11
      ON CUSTOMER11(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB12;
CREATE INDEX CUST_IDXB12
      ON CUSTOMER12(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB13;
CREATE INDEX CUST_IDXB13
      ON CUSTOMER13(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB14;
CREATE INDEX CUST_IDXB14
      ON CUSTOMER14(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;

```

```

DROP INDEX CUST_IDXB15;
CREATE INDEX CUST_IDXB15
      ON CUSTOMER15(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB16;
CREATE INDEX CUST_IDXB16
      ON CUSTOMER16(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB17;
CREATE INDEX CUST_IDXB17
      ON CUSTOMER17(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB18;
CREATE INDEX CUST_IDXB18
      ON CUSTOMER18(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB19;
CREATE INDEX CUST_IDXB19
      ON CUSTOMER19(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB20;
CREATE INDEX CUST_IDXB20
      ON CUSTOMER20(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB21;
CREATE INDEX CUST_IDXB21
      ON CUSTOMER21(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB22;
CREATE INDEX CUST_IDXB22
      ON CUSTOMER22(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB23;
CREATE INDEX CUST_IDXB23
      ON CUSTOMER23(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB24;
CREATE INDEX CUST_IDXB24
      ON CUSTOMER24(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB25;
CREATE INDEX CUST_IDXB25

```



```

connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB36;
CREATE INDEX ORDR_IDXB36
    ON ORDERS36(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB37;
CREATE INDEX ORDR_IDXB37
    ON ORDERS37(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB38;
CREATE INDEX ORDR_IDXB38
    ON ORDERS38(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB39;
CREATE INDEX ORDR_IDXB39
    ON ORDERS39(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB40;
CREATE INDEX ORDR_IDXB40
    ON ORDERS40(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB41;
CREATE INDEX ORDR_IDXB41
    ON ORDERS41(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB42;
CREATE INDEX ORDR_IDXB42
    ON ORDERS42(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB43;
CREATE INDEX ORDR_IDXB43
    ON ORDERS43(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB44;
CREATE INDEX ORDR_IDXB44
    ON ORDERS44(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB45;
CREATE INDEX ORDR_IDXB45
    ON ORDERS45(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;

```

```

DROP INDEX ORDR_IDXB46;
CREATE INDEX ORDR_IDXB46
    ON ORDERS46(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB47;
CREATE INDEX ORDR_IDXB47
    ON ORDERS47(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB48;
CREATE INDEX ORDR_IDXB48
    ON ORDERS48(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB49;
CREATE INDEX ORDR_IDXB49
    ON ORDERS49(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB50;
CREATE INDEX ORDR_IDXB50
    ON ORDERS50(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB51;
CREATE INDEX ORDR_IDXB51
    ON ORDERS51(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

```

DDL/CRTB_CUSTOMER.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER1;
CREATE TABLE CUSTOMER1
(
    C_ID          INTEGER      NOT NULL,
    C_STATE      CHAR(2)      NOT NULL,
    C_ZIP        CHAR(9)      NOT NULL,
    C_PHONE      CHAR(16)     NOT NULL,
    C_SINCE      BIGINT       NOT NULL,
    C_CREDIT_LIM BIGINT       NOT NULL,
    C_MIDDLE     CHAR(2)      NOT NULL,
    C_CREDIT     CHAR(2)      NOT NULL,
    C_DISCOUNT  INTEGER      NOT NULL,
    C_DATA       VARCHAR(500) NOT NULL,
    C_LAST       VARCHAR(16)  NOT NULL,
    C_FIRST      VARCHAR(16)  NOT NULL,
    C_STREET_1   VARCHAR(20)  NOT NULL,
    C_STREET_2   VARCHAR(20)  NOT NULL,
    C_CITY       VARCHAR(20)  NOT NULL,
    C_D_ID       SMALLINT     NOT NULL,
    C_W_ID       INTEGER      NOT NULL,
    C_DELIVERY_CNT INTEGER    NOT NULL,
    C_BALANCE    BIGINT       NOT NULL,

```

```

    C_YTD_PAYMENT BIGINT     NOT NULL,
    C_PAYMENT_CNT INTEGER     NOT NULL
)
IN ts_customer_001
INDEX IN is_customer_001
ORGANIZE BY KEY SEQUENCE (
    C_ID STARTING FROM 1 ENDING AT 3000,
    C_W_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER2;
CREATE TABLE CUSTOMER2
(
    C_ID          INTEGER      NOT NULL,
    C_STATE      CHAR(2)      NOT NULL,
    C_ZIP        CHAR(9)      NOT NULL,
    C_PHONE      CHAR(16)     NOT NULL,
    C_SINCE      BIGINT       NOT NULL,
    C_CREDIT_LIM BIGINT       NOT NULL,
    C_MIDDLE     CHAR(2)      NOT NULL,
    C_CREDIT     CHAR(2)      NOT NULL,
    C_DISCOUNT  INTEGER      NOT NULL,
    C_DATA       VARCHAR(500) NOT NULL,
    C_LAST       VARCHAR(16)  NOT NULL,
    C_FIRST      VARCHAR(16)  NOT NULL,
    C_STREET_1   VARCHAR(20)  NOT NULL,
    C_STREET_2   VARCHAR(20)  NOT NULL,
    C_CITY       VARCHAR(20)  NOT NULL,
    C_D_ID       SMALLINT     NOT NULL,
    C_W_ID       INTEGER      NOT NULL,
    C_DELIVERY_CNT INTEGER    NOT NULL,
    C_BALANCE    BIGINT       NOT NULL,
    C_YTD_PAYMENT BIGINT     NOT NULL,
    C_PAYMENT_CNT INTEGER     NOT NULL
)
IN ts_customer_002
INDEX IN is_customer_002
ORGANIZE BY KEY SEQUENCE (
    C_ID STARTING FROM 1 ENDING AT 3000,
    C_W_ID STARTING FROM 1601 ENDING AT
3200,
    C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER3;
CREATE TABLE CUSTOMER3
(
    C_ID          INTEGER      NOT NULL,
    C_STATE      CHAR(2)      NOT NULL,
    C_ZIP        CHAR(9)      NOT NULL,
    C_PHONE      CHAR(16)     NOT NULL,
    C_SINCE      BIGINT       NOT NULL,
    C_CREDIT_LIM BIGINT       NOT NULL,
    C_MIDDLE     CHAR(2)      NOT NULL,
    C_CREDIT     CHAR(2)      NOT NULL,
    C_DISCOUNT  INTEGER      NOT NULL,
    C_DATA       VARCHAR(500) NOT NULL,

```

```

C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_003
INDEX IN is_customer_003
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 3201 ENDING AT
4800,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER4;
CREATE TABLE CUSTOMER4
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_004
INDEX IN is_customer_004
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 4801 ENDING AT
6400,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER5;
CREATE TABLE CUSTOMER5
(

```

```

C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_005
INDEX IN is_customer_005
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 6401 ENDING AT
8000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER6;
CREATE TABLE CUSTOMER6
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_006
INDEX IN is_customer_006
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,

```

```

C_W_ID STARTING FROM 8001 ENDING AT
9600,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER7;
CREATE TABLE CUSTOMER7
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_007
INDEX IN is_customer_007
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 9601 ENDING AT
11200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER8;
CREATE TABLE CUSTOMER8
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,

```

```

C_W_ID    INTEGER    NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT    NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_008
INDEX IN is_customer_008
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 11201 ENDING AT
12800,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER9;
CREATE TABLE CUSTOMER9
(
  C_ID    INTEGER    NOT NULL,
  C_STATE CHAR(2)    NOT NULL,
  C_ZIP   CHAR(9)    NOT NULL,
  C_PHONE CHAR(16)   NOT NULL,
  C_SINCE BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT NOT NULL,
  C_MIDDLE CHAR(2)   NOT NULL,
  C_CREDIT CHAR(2)   NOT NULL,
  C_DISCOUNT INTEGER NOT NULL,
  C_DATA   VARCHAR(500) NOT NULL,
  C_LAST   VARCHAR(16) NOT NULL,
  C_FIRST  VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY   VARCHAR(20) NOT NULL,
  C_D_ID   SMALLINT  NOT NULL,
  C_W_ID   INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT     NOT NULL,
  C_YTD_PAYMENT BIGINT NOT NULL,
  C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_009
INDEX IN is_customer_009
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 12801 ENDING AT
14400,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER10;
CREATE TABLE CUSTOMER10
(
  C_ID    INTEGER    NOT NULL,
  C_STATE CHAR(2)    NOT NULL,
  C_ZIP   CHAR(9)    NOT NULL,
  C_PHONE CHAR(16)   NOT NULL,
  C_SINCE BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT NOT NULL,

```

```

C_MIDDLE CHAR(2)    NOT NULL,
C_CREDIT CHAR(2)    NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA   VARCHAR(500) NOT NULL,
C_LAST   VARCHAR(16) NOT NULL,
C_FIRST  VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY   VARCHAR(20) NOT NULL,
C_D_ID   SMALLINT  NOT NULL,
C_W_ID   INTEGER    NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT     NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_010
INDEX IN is_customer_010
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 14401 ENDING AT
16000,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER11;
CREATE TABLE CUSTOMER11
(
  C_ID    INTEGER    NOT NULL,
  C_STATE CHAR(2)    NOT NULL,
  C_ZIP   CHAR(9)    NOT NULL,
  C_PHONE CHAR(16)   NOT NULL,
  C_SINCE BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT NOT NULL,
  C_MIDDLE CHAR(2)   NOT NULL,
  C_CREDIT CHAR(2)   NOT NULL,
  C_DISCOUNT INTEGER NOT NULL,
  C_DATA   VARCHAR(500) NOT NULL,
  C_LAST   VARCHAR(16) NOT NULL,
  C_FIRST  VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY   VARCHAR(20) NOT NULL,
  C_D_ID   SMALLINT  NOT NULL,
  C_W_ID   INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT     NOT NULL,
  C_YTD_PAYMENT BIGINT NOT NULL,
  C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_011
INDEX IN is_customer_011
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 16001 ENDING AT
17600,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER12;
CREATE TABLE CUSTOMER12
(
  C_ID    INTEGER    NOT NULL,
  C_STATE CHAR(2)    NOT NULL,
  C_ZIP   CHAR(9)    NOT NULL,
  C_PHONE CHAR(16)   NOT NULL,
  C_SINCE BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT NOT NULL,
  C_MIDDLE CHAR(2)   NOT NULL,
  C_CREDIT CHAR(2)   NOT NULL,
  C_DISCOUNT INTEGER NOT NULL,
  C_DATA   VARCHAR(500) NOT NULL,
  C_LAST   VARCHAR(16) NOT NULL,
  C_FIRST  VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY   VARCHAR(20) NOT NULL,
  C_D_ID   SMALLINT  NOT NULL,
  C_W_ID   INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT     NOT NULL,
  C_YTD_PAYMENT BIGINT NOT NULL,
  C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_012
INDEX IN is_customer_012
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 17601 ENDING AT
19200,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER13;
CREATE TABLE CUSTOMER13
(
  C_ID    INTEGER    NOT NULL,
  C_STATE CHAR(2)    NOT NULL,
  C_ZIP   CHAR(9)    NOT NULL,
  C_PHONE CHAR(16)   NOT NULL,
  C_SINCE BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT NOT NULL,
  C_MIDDLE CHAR(2)   NOT NULL,
  C_CREDIT CHAR(2)   NOT NULL,
  C_DISCOUNT INTEGER NOT NULL,
  C_DATA   VARCHAR(500) NOT NULL,
  C_LAST   VARCHAR(16) NOT NULL,
  C_FIRST  VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY   VARCHAR(20) NOT NULL,
  C_D_ID   SMALLINT  NOT NULL,
  C_W_ID   INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT     NOT NULL,
  C_YTD_PAYMENT BIGINT NOT NULL,
  C_PAYMENT_CNT INTEGER NOT NULL
)

```

```

IN ts_customer_013
INDEX IN is_customer_013
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 19201 ENDING AT
20800,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER14;
CREATE TABLE CUSTOMER14
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_014
INDEX IN is_customer_014
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 20801 ENDING AT
22400,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER15;
CREATE TABLE CUSTOMER15
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,

```

```

C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_015
INDEX IN is_customer_015
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 22401 ENDING AT
24000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER16;
CREATE TABLE CUSTOMER16
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_016
INDEX IN is_customer_016
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 24001 ENDING AT
25600,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER17;
CREATE TABLE CUSTOMER17
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,

```

```

C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_017
INDEX IN is_customer_017
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 25601 ENDING AT
27200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER18;
CREATE TABLE CUSTOMER18
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_018
INDEX IN is_customer_018
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 27201 ENDING AT
28800,

```

```

        C_D_ID STARTING FROM 1 ENDING AT 10
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER19;
CREATE TABLE CUSTOMER19
(
    C_ID          INTEGER      NOT NULL,
    C_STATE       CHAR(2)      NOT NULL,
    C_ZIP         CHAR(9)      NOT NULL,
    C_PHONE       CHAR(16)     NOT NULL,
    C_SINCE       BIGINT       NOT NULL,
    C_CREDIT_LIM  BIGINT       NOT NULL,
    C_MIDDLE      CHAR(2)      NOT NULL,
    C_CREDIT      CHAR(2)      NOT NULL,
    C_DISCOUNT   INTEGER      NOT NULL,
    C_DATA        VARCHAR(500) NOT NULL,
    C_LAST        VARCHAR(16)  NOT NULL,
    C_FIRST       VARCHAR(16)  NOT NULL,
    C_STREET_1    VARCHAR(20)  NOT NULL,
    C_STREET_2    VARCHAR(20)  NOT NULL,
    C_CITY        VARCHAR(20)  NOT NULL,
    C_D_ID        SMALLINT     NOT NULL,
    C_W_ID        INTEGER      NOT NULL,
    C_DELIVERY_CNT INTEGER     NOT NULL,
    C_BALANCE     BIGINT       NOT NULL,
    C_YTD_PAYMENT BIGINT       NOT NULL,
    C_PAYMENT_CNT INTEGER      NOT NULL
)
IN ts_customer_019
INDEX IN is_customer_019
ORGANIZE BY KEY SEQUENCE (
    C_ID STARTING FROM 1 ENDING AT 3000,
    C_W_ID STARTING FROM 28801 ENDING AT
30400,
    C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER20;
CREATE TABLE CUSTOMER20
(
    C_ID          INTEGER      NOT NULL,
    C_STATE       CHAR(2)      NOT NULL,
    C_ZIP         CHAR(9)      NOT NULL,
    C_PHONE       CHAR(16)     NOT NULL,
    C_SINCE       BIGINT       NOT NULL,
    C_CREDIT_LIM  BIGINT       NOT NULL,
    C_MIDDLE      CHAR(2)      NOT NULL,
    C_CREDIT      CHAR(2)      NOT NULL,
    C_DISCOUNT   INTEGER      NOT NULL,
    C_DATA        VARCHAR(500) NOT NULL,
    C_LAST        VARCHAR(16)  NOT NULL,
    C_FIRST       VARCHAR(16)  NOT NULL,
    C_STREET_1    VARCHAR(20)  NOT NULL,
    C_STREET_2    VARCHAR(20)  NOT NULL,
    C_CITY        VARCHAR(20)  NOT NULL,
    C_D_ID        SMALLINT     NOT NULL,
    C_W_ID        INTEGER      NOT NULL,
    C_DELIVERY_CNT INTEGER     NOT NULL,

```

```

    C_BALANCE     BIGINT       NOT NULL,
    C_YTD_PAYMENT BIGINT       NOT NULL,
    C_PAYMENT_CNT INTEGER      NOT NULL
)
IN ts_customer_020
INDEX IN is_customer_020
ORGANIZE BY KEY SEQUENCE (
    C_ID STARTING FROM 1 ENDING AT 3000,
    C_W_ID STARTING FROM 30401 ENDING AT
32000,
    C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER21;
CREATE TABLE CUSTOMER21
(
    C_ID          INTEGER      NOT NULL,
    C_STATE       CHAR(2)      NOT NULL,
    C_ZIP         CHAR(9)      NOT NULL,
    C_PHONE       CHAR(16)     NOT NULL,
    C_SINCE       BIGINT       NOT NULL,
    C_CREDIT_LIM  BIGINT       NOT NULL,
    C_MIDDLE      CHAR(2)      NOT NULL,
    C_CREDIT      CHAR(2)      NOT NULL,
    C_DISCOUNT   INTEGER      NOT NULL,
    C_DATA        VARCHAR(500) NOT NULL,
    C_LAST        VARCHAR(16)  NOT NULL,
    C_FIRST       VARCHAR(16)  NOT NULL,
    C_STREET_1    VARCHAR(20)  NOT NULL,
    C_STREET_2    VARCHAR(20)  NOT NULL,
    C_CITY        VARCHAR(20)  NOT NULL,
    C_D_ID        SMALLINT     NOT NULL,
    C_W_ID        INTEGER      NOT NULL,
    C_DELIVERY_CNT INTEGER     NOT NULL,
    C_BALANCE     BIGINT       NOT NULL,
    C_YTD_PAYMENT BIGINT       NOT NULL,
    C_PAYMENT_CNT INTEGER      NOT NULL
)
IN ts_customer_021
INDEX IN is_customer_021
ORGANIZE BY KEY SEQUENCE (
    C_ID STARTING FROM 1 ENDING AT 3000,
    C_W_ID STARTING FROM 32001 ENDING AT
33600,
    C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER22;
CREATE TABLE CUSTOMER22
(
    C_ID          INTEGER      NOT NULL,
    C_STATE       CHAR(2)      NOT NULL,
    C_ZIP         CHAR(9)      NOT NULL,
    C_PHONE       CHAR(16)     NOT NULL,
    C_SINCE       BIGINT       NOT NULL,
    C_CREDIT_LIM  BIGINT       NOT NULL,
    C_MIDDLE      CHAR(2)      NOT NULL,
    C_CREDIT      CHAR(2)      NOT NULL,

```

```

    C_DISCOUNT   INTEGER      NOT NULL,
    C_DATA        VARCHAR(500) NOT NULL,
    C_LAST        VARCHAR(16)  NOT NULL,
    C_FIRST       VARCHAR(16)  NOT NULL,
    C_STREET_1    VARCHAR(20)  NOT NULL,
    C_STREET_2    VARCHAR(20)  NOT NULL,
    C_CITY        VARCHAR(20)  NOT NULL,
    C_D_ID        SMALLINT     NOT NULL,
    C_W_ID        INTEGER      NOT NULL,
    C_DELIVERY_CNT INTEGER     NOT NULL,
    C_BALANCE     BIGINT       NOT NULL,
    C_YTD_PAYMENT BIGINT       NOT NULL,
    C_PAYMENT_CNT INTEGER      NOT NULL
)
IN ts_customer_022
INDEX IN is_customer_022
ORGANIZE BY KEY SEQUENCE (
    C_ID STARTING FROM 1 ENDING AT 3000,
    C_W_ID STARTING FROM 33601 ENDING AT
35200,
    C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER23;
CREATE TABLE CUSTOMER23
(
    C_ID          INTEGER      NOT NULL,
    C_STATE       CHAR(2)      NOT NULL,
    C_ZIP         CHAR(9)      NOT NULL,
    C_PHONE       CHAR(16)     NOT NULL,
    C_SINCE       BIGINT       NOT NULL,
    C_CREDIT_LIM  BIGINT       NOT NULL,
    C_MIDDLE      CHAR(2)      NOT NULL,
    C_CREDIT      CHAR(2)      NOT NULL,
    C_DISCOUNT   INTEGER      NOT NULL,
    C_DATA        VARCHAR(500) NOT NULL,
    C_LAST        VARCHAR(16)  NOT NULL,
    C_FIRST       VARCHAR(16)  NOT NULL,
    C_STREET_1    VARCHAR(20)  NOT NULL,
    C_STREET_2    VARCHAR(20)  NOT NULL,
    C_CITY        VARCHAR(20)  NOT NULL,
    C_D_ID        SMALLINT     NOT NULL,
    C_W_ID        INTEGER      NOT NULL,
    C_DELIVERY_CNT INTEGER     NOT NULL,
    C_BALANCE     BIGINT       NOT NULL,
    C_YTD_PAYMENT BIGINT       NOT NULL,
    C_PAYMENT_CNT INTEGER      NOT NULL
)
IN ts_customer_023
INDEX IN is_customer_023
ORGANIZE BY KEY SEQUENCE (
    C_ID STARTING FROM 1 ENDING AT 3000,
    C_W_ID STARTING FROM 35201 ENDING AT
36800,
    C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER24;

```



```

CREATE TABLE CUSTOMER24
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_024
INDEX IN is_customer_024
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 36801 ENDING AT
38400,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER25;
CREATE TABLE CUSTOMER25
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_025
INDEX IN is_customer_025

```

```

ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 38401 ENDING AT
40000,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER26;
CREATE TABLE CUSTOMER26
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_026
INDEX IN is_customer_026
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 40001 ENDING AT
41600,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER27;
CREATE TABLE CUSTOMER27
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,

```

```

  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_027
INDEX IN is_customer_027
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 41601 ENDING AT
43200,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER28;
CREATE TABLE CUSTOMER28
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_028
INDEX IN is_customer_028
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 43201 ENDING AT
44800,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER29;
CREATE TABLE CUSTOMER29
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,

```

```

C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_029
INDEX IN is_customer_029
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 44801 ENDING AT
46400,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER30;
CREATE TABLE CUSTOMER30
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_030
INDEX IN is_customer_030
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 46401 ENDING AT
48000,
C_D_ID STARTING FROM 1 ENDING AT 10
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER31;
CREATE TABLE CUSTOMER31
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_031
INDEX IN is_customer_031
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 48001 ENDING AT
49600,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER32;
CREATE TABLE CUSTOMER32
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,

```

```

C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_032
INDEX IN is_customer_032
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 49601 ENDING AT
51200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER33;
CREATE TABLE CUSTOMER33
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_033
INDEX IN is_customer_033
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 51201 ENDING AT
52800,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER34;
CREATE TABLE CUSTOMER34
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,

```

```

C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_034
INDEX IN is_customer_034
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 52801 ENDING AT
54400,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER35;
CREATE TABLE CUSTOMER35
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_035
INDEX IN is_customer_035
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 54401 ENDING AT
56000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER36;
CREATE TABLE CUSTOMER36
(

```

```

C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_036
INDEX IN is_customer_036
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 56001 ENDING AT
57600,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER37;
CREATE TABLE CUSTOMER37
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_037
INDEX IN is_customer_037
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,

```

```

C_W_ID STARTING FROM 57601 ENDING AT
59200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER38;
CREATE TABLE CUSTOMER38
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_038
INDEX IN is_customer_038
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 59201 ENDING AT
60800,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER39;
CREATE TABLE CUSTOMER39
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,

```

```

C_W_ID    INTEGER    NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT    NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_039
INDEX IN is_customer_039
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 60801 ENDING AT
62400,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER40;
CREATE TABLE CUSTOMER40
(
C_ID    INTEGER    NOT NULL,
C_STATE CHAR(2)    NOT NULL,
C_ZIP   CHAR(9)    NOT NULL,
C_PHONE CHAR(16)   NOT NULL,
C_SINCE BIGINT     NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2)   NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA   VARCHAR(500) NOT NULL,
C_LAST   VARCHAR(16) NOT NULL,
C_FIRST  VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY   VARCHAR(20) NOT NULL,
C_D_ID   SMALLINT  NOT NULL,
C_W_ID   INTEGER    NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT     NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_040
INDEX IN is_customer_040
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 62401 ENDING AT
64000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER41;
CREATE TABLE CUSTOMER41
(
C_ID    INTEGER    NOT NULL,
C_STATE CHAR(2)    NOT NULL,
C_ZIP   CHAR(9)    NOT NULL,
C_PHONE CHAR(16)   NOT NULL,
C_SINCE BIGINT     NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,

```

```

C_MIDDLE CHAR(2)    NOT NULL,
C_CREDIT CHAR(2)    NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA   VARCHAR(500) NOT NULL,
C_LAST   VARCHAR(16) NOT NULL,
C_FIRST  VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY   VARCHAR(20) NOT NULL,
C_D_ID   SMALLINT  NOT NULL,
C_W_ID   INTEGER    NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT     NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_041
INDEX IN is_customer_041
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 64001 ENDING AT
65600,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER42;
CREATE TABLE CUSTOMER42
(
C_ID    INTEGER    NOT NULL,
C_STATE CHAR(2)    NOT NULL,
C_ZIP   CHAR(9)    NOT NULL,
C_PHONE CHAR(16)   NOT NULL,
C_SINCE BIGINT     NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2)   NOT NULL,
C_CREDIT CHAR(2)   NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA   VARCHAR(500) NOT NULL,
C_LAST   VARCHAR(16) NOT NULL,
C_FIRST  VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY   VARCHAR(20) NOT NULL,
C_D_ID   SMALLINT  NOT NULL,
C_W_ID   INTEGER    NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT     NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_042
INDEX IN is_customer_042
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 65601 ENDING AT
67200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER43;
CREATE TABLE CUSTOMER43
(
C_ID    INTEGER    NOT NULL,
C_STATE CHAR(2)    NOT NULL,
C_ZIP   CHAR(9)    NOT NULL,
C_PHONE CHAR(16)   NOT NULL,
C_SINCE BIGINT     NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2)   NOT NULL,
C_CREDIT CHAR(2)   NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA   VARCHAR(500) NOT NULL,
C_LAST   VARCHAR(16) NOT NULL,
C_FIRST  VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY   VARCHAR(20) NOT NULL,
C_D_ID   SMALLINT  NOT NULL,
C_W_ID   INTEGER    NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT     NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_043
INDEX IN is_customer_043
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 67201 ENDING AT
68800,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER44;
CREATE TABLE CUSTOMER44
(
C_ID    INTEGER    NOT NULL,
C_STATE CHAR(2)    NOT NULL,
C_ZIP   CHAR(9)    NOT NULL,
C_PHONE CHAR(16)   NOT NULL,
C_SINCE BIGINT     NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2)   NOT NULL,
C_CREDIT CHAR(2)   NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA   VARCHAR(500) NOT NULL,
C_LAST   VARCHAR(16) NOT NULL,
C_FIRST  VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY   VARCHAR(20) NOT NULL,
C_D_ID   SMALLINT  NOT NULL,
C_W_ID   INTEGER    NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT     NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)

```

```

IN ts_customer_044
INDEX IN is_customer_044
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 68801 ENDING AT
70400,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER45;
CREATE TABLE CUSTOMER45
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT    NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER    NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_045
INDEX IN is_customer_045
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 70401 ENDING AT
72000,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER46;
CREATE TABLE CUSTOMER46
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT    NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER    NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,

```

```

  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_046
INDEX IN is_customer_046
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 72001 ENDING AT
73600,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER47;
CREATE TABLE CUSTOMER47
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT    NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER    NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_047
INDEX IN is_customer_047
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 73601 ENDING AT
75200,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER48;
CREATE TABLE CUSTOMER48
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,

```

```

  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT    NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER    NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_048
INDEX IN is_customer_048
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 75201 ENDING AT
76800,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER49;
CREATE TABLE CUSTOMER49
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT    NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER    NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_049
INDEX IN is_customer_049
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 76801 ENDING AT
78400,

```

```

C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER50;
CREATE TABLE CUSTOMER50
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_050
INDEX IN is_customer_050
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 78401 ENDING AT
80000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER51;
CREATE TABLE CUSTOMER51
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,

```

```

C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_051
INDEX IN is_customer_051
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 80001 ENDING AT
81600,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

DDL/CRTB DISTRICT.ddl

connect to TPCC in share mode;
DROP TABLE DISTRICT1;
CREATE TABLE DISTRICT1
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_001
INDEX IN ts_dist_001
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 1 ENDING AT 1600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT2;
CREATE TABLE DISTRICT2
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_002
INDEX IN ts_dist_002
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,

```

```

D_W_ID STARTING FROM 1601 ENDING AT
3200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT3;
CREATE TABLE DISTRICT3
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_003
INDEX IN ts_dist_003
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 3201 ENDING AT
4800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT4;
CREATE TABLE DISTRICT4
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_004
INDEX IN ts_dist_004
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 4801 ENDING AT
6400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT5;
CREATE TABLE DISTRICT5
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,

```

```

D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_005
INDEX IN ts_dist_005
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 6401 ENDING AT
8000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT6;
CREATE TABLE DISTRICT6
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_006
INDEX IN ts_dist_006
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 8001 ENDING AT
9600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT7;
CREATE TABLE DISTRICT7
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_007
INDEX IN ts_dist_007
ORGANIZE BY KEY SEQUENCE (

```

```

D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 9601 ENDING AT
11200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT8;
CREATE TABLE DISTRICT8
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_008
INDEX IN ts_dist_008
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 11201 ENDING AT
12800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT9;
CREATE TABLE DISTRICT9
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_009
INDEX IN ts_dist_009
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 12801 ENDING AT
14400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT10;
CREATE TABLE DISTRICT10
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,

```

```

D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_010
INDEX IN ts_dist_010
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 14401 ENDING AT
16000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT11;
CREATE TABLE DISTRICT11
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_011
INDEX IN ts_dist_011
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 16001 ENDING AT
17600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT12;
CREATE TABLE DISTRICT12
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_012
INDEX IN ts_dist_012

```

```

ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 17601 ENDING AT
19200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT13;
CREATE TABLE DISTRICT13
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_013
INDEX IN ts_dist_013
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 19201 ENDING AT
20800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT14;
CREATE TABLE DISTRICT14
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_014
INDEX IN ts_dist_014
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 20801 ENDING AT
22400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT15;
CREATE TABLE DISTRICT15
(
  D_NEXT_O_ID INTEGER NOT NULL,

```

```

  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_015
INDEX IN ts_dist_015
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 22401 ENDING AT
24000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT16;
CREATE TABLE DISTRICT16
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_016
INDEX IN ts_dist_016
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 24001 ENDING AT
25600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT17;
CREATE TABLE DISTRICT17
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_017

```

```

INDEX IN ts_dist_017
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 25601 ENDING AT
27200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT18;
CREATE TABLE DISTRICT18
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_018
INDEX IN ts_dist_018
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 27201 ENDING AT
28800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT19;
CREATE TABLE DISTRICT19
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_019
INDEX IN ts_dist_019
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 28801 ENDING AT
30400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT20;
CREATE TABLE DISTRICT20
(

```



```

D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_020
INDEX IN ts_dist_020
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 30401 ENDING AT
32000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT21;
CREATE TABLE DISTRICT21
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_021
INDEX IN ts_dist_021
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 32001 ENDING AT
33600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT22;
CREATE TABLE DISTRICT22
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)

```

```

IN ts_dist_022
INDEX IN ts_dist_022
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 33601 ENDING AT
35200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT23;
CREATE TABLE DISTRICT23
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_023
INDEX IN ts_dist_023
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 35201 ENDING AT
36800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT24;
CREATE TABLE DISTRICT24
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_024
INDEX IN ts_dist_024
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 36801 ENDING AT
38400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT25;
CREATE TABLE DISTRICT25

```

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_025
INDEX IN ts_dist_025
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 38401 ENDING AT
40000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT26;
CREATE TABLE DISTRICT26
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_026
INDEX IN ts_dist_026
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 40001 ENDING AT
41600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT27;
CREATE TABLE DISTRICT27
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)

```

```

)
IN ts_dist_027
INDEX IN ts_dist_027
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 41601 ENDING AT
43200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT28;
CREATE TABLE DISTRICT28
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_028
INDEX IN ts_dist_028
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 43201 ENDING AT
44800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT29;
CREATE TABLE DISTRICT29
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_029
INDEX IN ts_dist_029
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 44801 ENDING AT
46400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT30;

```

```

CREATE TABLE DISTRICT30
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_030
INDEX IN ts_dist_030
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 46401 ENDING AT
48000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT31;
CREATE TABLE DISTRICT31
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_031
INDEX IN ts_dist_031
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 48001 ENDING AT
49600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT32;
CREATE TABLE DISTRICT32
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,

```

```

  D_W_ID INTEGER NOT NULL
)
IN ts_dist_032
INDEX IN ts_dist_032
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 49601 ENDING AT
51200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT33;
CREATE TABLE DISTRICT33
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_033
INDEX IN ts_dist_033
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 51201 ENDING AT
52800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT34;
CREATE TABLE DISTRICT34
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_034
INDEX IN ts_dist_034
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 52801 ENDING AT
54400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;

```

```

DROP TABLE DISTRICT35;
CREATE TABLE DISTRICT35
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_035
INDEX IN ts_dist_035
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 54401 ENDING AT
56000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT36;
CREATE TABLE DISTRICT36
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_036
INDEX IN ts_dist_036
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 56001 ENDING AT
57600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT37;
CREATE TABLE DISTRICT37
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,

```

```

  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_037
INDEX IN ts_dist_037
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 57601 ENDING AT
59200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT38;
CREATE TABLE DISTRICT38
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_038
INDEX IN ts_dist_038
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 59201 ENDING AT
60800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT39;
CREATE TABLE DISTRICT39
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_039
INDEX IN ts_dist_039
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 60801 ENDING AT
62400
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE DISTRICT40;
CREATE TABLE DISTRICT40
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_040
INDEX IN ts_dist_040
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 62401 ENDING AT
64000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT41;
CREATE TABLE DISTRICT41
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_041
INDEX IN ts_dist_041
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 64001 ENDING AT
65600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT42;
CREATE TABLE DISTRICT42
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,

```

```

D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_042
INDEX IN ts_dist_042
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 65601 ENDING AT
67200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT43;
CREATE TABLE DISTRICT43
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_043
INDEX IN ts_dist_043
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 67201 ENDING AT
68800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT44;
CREATE TABLE DISTRICT44
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_044
INDEX IN ts_dist_044
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 68801 ENDING AT
70400
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT45;
CREATE TABLE DISTRICT45
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_045
INDEX IN ts_dist_045
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 70401 ENDING AT
72000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT46;
CREATE TABLE DISTRICT46
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_046
INDEX IN ts_dist_046
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 72001 ENDING AT
73600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT47;
CREATE TABLE DISTRICT47
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,

```

```

D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_047
INDEX IN ts_dist_047
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 73601 ENDING AT
75200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT48;
CREATE TABLE DISTRICT48
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_048
INDEX IN ts_dist_048
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 75201 ENDING AT
76800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT49;
CREATE TABLE DISTRICT49
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_049
INDEX IN ts_dist_049
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 76801 ENDING AT
78400
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT50;
CREATE TABLE DISTRICT50
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_050
INDEX IN ts_dist_050
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 78401 ENDING AT
80000
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT51;
CREATE TABLE DISTRICT51
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_051
INDEX IN ts_dist_051
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 80001 ENDING AT
81600
)

```

```

ALLOW OVERFLOW;
connect reset;

```

DDL/CRTB HISTORY.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY1;
CREATE TABLE HISTORY1
(
  H_C_ID INTEGER NOT NULL,
  H_C_D_ID SMALLINT NOT NULL,

```

```

  H_C_W_ID INTEGER NOT NULL,
  H_D_ID SMALLINT NOT NULL,
  H_W_ID INTEGER NOT NULL,
  H_DATE BIGINT NOT NULL,
  H_AMOUNT INTEGER NOT NULL,
  H_DATA CHAR(24) NOT NULL
)
IN ts_history_001
INDEX IN ts_history_001;

```

```

ALTER TABLE HISTORY1 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY2;
CREATE TABLE HISTORY2
(
  H_C_ID INTEGER NOT NULL,
  H_C_D_ID SMALLINT NOT NULL,
  H_C_W_ID INTEGER NOT NULL,
  H_D_ID SMALLINT NOT NULL,
  H_W_ID INTEGER NOT NULL,
  H_DATE BIGINT NOT NULL,
  H_AMOUNT INTEGER NOT NULL,
  H_DATA CHAR(24) NOT NULL
)
IN ts_history_002
INDEX IN ts_history_002;

```

```

ALTER TABLE HISTORY2 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY3;
CREATE TABLE HISTORY3
(
  H_C_ID INTEGER NOT NULL,
  H_C_D_ID SMALLINT NOT NULL,
  H_C_W_ID INTEGER NOT NULL,
  H_D_ID SMALLINT NOT NULL,
  H_W_ID INTEGER NOT NULL,
  H_DATE BIGINT NOT NULL,
  H_AMOUNT INTEGER NOT NULL,
  H_DATA CHAR(24) NOT NULL
)
IN ts_history_003
INDEX IN ts_history_003;

```

```

ALTER TABLE HISTORY3 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY4;
CREATE TABLE HISTORY4
(
  H_C_ID INTEGER NOT NULL,
  H_C_D_ID SMALLINT NOT NULL,
  H_C_W_ID INTEGER NOT NULL,
  H_D_ID SMALLINT NOT NULL,
  H_W_ID INTEGER NOT NULL,
  H_DATE BIGINT NOT NULL,
  H_AMOUNT INTEGER NOT NULL,
  H_DATA CHAR(24) NOT NULL
)
IN ts_history_004
INDEX IN ts_history_004;

```

```

ALTER TABLE HISTORY4 APPEND ON;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE HISTORY5;
CREATE TABLE HISTORY5
(
  H_C_ID INTEGER NOT NULL,
  H_C_D_ID SMALLINT NOT NULL,
  H_C_W_ID INTEGER NOT NULL,
  H_D_ID SMALLINT NOT NULL,
  H_W_ID INTEGER NOT NULL,
  H_DATE BIGINT NOT NULL,
  H_AMOUNT INTEGER NOT NULL,
  H_DATA CHAR(24) NOT NULL
)
IN ts_history_005
INDEX IN ts_history_005;

```

```

ALTER TABLE HISTORY5 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY6;
CREATE TABLE HISTORY6
(
  H_C_ID INTEGER NOT NULL,
  H_C_D_ID SMALLINT NOT NULL,
  H_C_W_ID INTEGER NOT NULL,
  H_D_ID SMALLINT NOT NULL,
  H_W_ID INTEGER NOT NULL,
  H_DATE BIGINT NOT NULL,
  H_AMOUNT INTEGER NOT NULL,
  H_DATA CHAR(24) NOT NULL
)
IN ts_history_006
INDEX IN ts_history_006;

```

```

ALTER TABLE HISTORY6 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY7;
CREATE TABLE HISTORY7
(
  H_C_ID INTEGER NOT NULL,
  H_C_D_ID SMALLINT NOT NULL,
  H_C_W_ID INTEGER NOT NULL,
  H_D_ID SMALLINT NOT NULL,
  H_W_ID INTEGER NOT NULL,
  H_DATE BIGINT NOT NULL,
  H_AMOUNT INTEGER NOT NULL,
  H_DATA CHAR(24) NOT NULL
)
IN ts_history_007
INDEX IN ts_history_007;

```

```

ALTER TABLE HISTORY7 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY8;
CREATE TABLE HISTORY8
(
  H_C_ID INTEGER NOT NULL,
  H_C_D_ID SMALLINT NOT NULL,
  H_C_W_ID INTEGER NOT NULL,
  H_D_ID SMALLINT NOT NULL,
  H_W_ID INTEGER NOT NULL,
  H_DATE BIGINT NOT NULL,
  H_AMOUNT INTEGER NOT NULL,

```

```

        H_DATA    CHAR(24) NOT NULL
    )
    IN ts_history_008
    INDEX IN ts_history_008;
ALTER TABLE HISTORY8 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY9;
CREATE TABLE HISTORY9
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT NOT NULL,
    H_AMOUNT  INTEGER NOT NULL,
    H_DATA    CHAR(24) NOT NULL
)
IN ts_history_009
INDEX IN ts_history_009;
ALTER TABLE HISTORY9 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY10;
CREATE TABLE HISTORY10
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT NOT NULL,
    H_AMOUNT  INTEGER NOT NULL,
    H_DATA    CHAR(24) NOT NULL
)
IN ts_history_010
INDEX IN ts_history_010;
ALTER TABLE HISTORY10 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY11;
CREATE TABLE HISTORY11
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT NOT NULL,
    H_AMOUNT  INTEGER NOT NULL,
    H_DATA    CHAR(24) NOT NULL
)
IN ts_history_011
INDEX IN ts_history_011;
ALTER TABLE HISTORY11 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY12;
CREATE TABLE HISTORY12
(
    H_C_ID    INTEGER NOT NULL,

```

```

        H_C_D_ID  SMALLINT NOT NULL,
        H_C_W_ID  INTEGER NOT NULL,
        H_D_ID    SMALLINT NOT NULL,
        H_W_ID    INTEGER NOT NULL,
        H_DATE    BIGINT NOT NULL,
        H_AMOUNT  INTEGER NOT NULL,
        H_DATA    CHAR(24) NOT NULL
    )
    IN ts_history_012
    INDEX IN ts_history_012;
ALTER TABLE HISTORY12 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY13;
CREATE TABLE HISTORY13
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT NOT NULL,
    H_AMOUNT  INTEGER NOT NULL,
    H_DATA    CHAR(24) NOT NULL
)
IN ts_history_013
INDEX IN ts_history_013;
ALTER TABLE HISTORY13 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY14;
CREATE TABLE HISTORY14
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT NOT NULL,
    H_AMOUNT  INTEGER NOT NULL,
    H_DATA    CHAR(24) NOT NULL
)
IN ts_history_014
INDEX IN ts_history_014;
ALTER TABLE HISTORY14 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY15;
CREATE TABLE HISTORY15
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT NOT NULL,
    H_AMOUNT  INTEGER NOT NULL,
    H_DATA    CHAR(24) NOT NULL
)
IN ts_history_015
INDEX IN ts_history_015;
ALTER TABLE HISTORY15 APPEND ON;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY16;
CREATE TABLE HISTORY16
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT NOT NULL,
    H_AMOUNT  INTEGER NOT NULL,
    H_DATA    CHAR(24) NOT NULL
)
IN ts_history_016
INDEX IN ts_history_016;
ALTER TABLE HISTORY16 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY17;
CREATE TABLE HISTORY17
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT NOT NULL,
    H_AMOUNT  INTEGER NOT NULL,
    H_DATA    CHAR(24) NOT NULL
)
IN ts_history_017
INDEX IN ts_history_017;
ALTER TABLE HISTORY17 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY18;
CREATE TABLE HISTORY18
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT NOT NULL,
    H_AMOUNT  INTEGER NOT NULL,
    H_DATA    CHAR(24) NOT NULL
)
IN ts_history_018
INDEX IN ts_history_018;
ALTER TABLE HISTORY18 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY19;
CREATE TABLE HISTORY19
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT NOT NULL,

```

```

        H_AMOUNT INTEGER NOT NULL,
        H_DATA CHAR(24) NOT NULL
    )
    IN ts_history_019
    INDEX IN ts_history_019;
ALTER TABLE HISTORY19 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY20;
CREATE TABLE HISTORY20
(
    H_C_ID INTEGER NOT NULL,
    H_C_D_ID SMALLINT NOT NULL,
    H_C_W_ID INTEGER NOT NULL,
    H_D_ID SMALLINT NOT NULL,
    H_W_ID INTEGER NOT NULL,
    H_DATE BIGINT NOT NULL,
    H_AMOUNT INTEGER NOT NULL,
    H_DATA CHAR(24) NOT NULL
)
    IN ts_history_020
    INDEX IN ts_history_020;
ALTER TABLE HISTORY20 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY21;
CREATE TABLE HISTORY21
(
    H_C_ID INTEGER NOT NULL,
    H_C_D_ID SMALLINT NOT NULL,
    H_C_W_ID INTEGER NOT NULL,
    H_D_ID SMALLINT NOT NULL,
    H_W_ID INTEGER NOT NULL,
    H_DATE BIGINT NOT NULL,
    H_AMOUNT INTEGER NOT NULL,
    H_DATA CHAR(24) NOT NULL
)
    IN ts_history_021
    INDEX IN ts_history_021;
ALTER TABLE HISTORY21 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY22;
CREATE TABLE HISTORY22
(
    H_C_ID INTEGER NOT NULL,
    H_C_D_ID SMALLINT NOT NULL,
    H_C_W_ID INTEGER NOT NULL,
    H_D_ID SMALLINT NOT NULL,
    H_W_ID INTEGER NOT NULL,
    H_DATE BIGINT NOT NULL,
    H_AMOUNT INTEGER NOT NULL,
    H_DATA CHAR(24) NOT NULL
)
    IN ts_history_022
    INDEX IN ts_history_022;
ALTER TABLE HISTORY22 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY23;
CREATE TABLE HISTORY23
(

```

```

        H_C_ID INTEGER NOT NULL,
        H_C_D_ID SMALLINT NOT NULL,
        H_C_W_ID INTEGER NOT NULL,
        H_D_ID SMALLINT NOT NULL,
        H_W_ID INTEGER NOT NULL,
        H_DATE BIGINT NOT NULL,
        H_AMOUNT INTEGER NOT NULL,
        H_DATA CHAR(24) NOT NULL
    )
    IN ts_history_023
    INDEX IN ts_history_023;
ALTER TABLE HISTORY23 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY24;
CREATE TABLE HISTORY24
(
    H_C_ID INTEGER NOT NULL,
    H_C_D_ID SMALLINT NOT NULL,
    H_C_W_ID INTEGER NOT NULL,
    H_D_ID SMALLINT NOT NULL,
    H_W_ID INTEGER NOT NULL,
    H_DATE BIGINT NOT NULL,
    H_AMOUNT INTEGER NOT NULL,
    H_DATA CHAR(24) NOT NULL
)
    IN ts_history_024
    INDEX IN ts_history_024;
ALTER TABLE HISTORY24 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY25;
CREATE TABLE HISTORY25
(
    H_C_ID INTEGER NOT NULL,
    H_C_D_ID SMALLINT NOT NULL,
    H_C_W_ID INTEGER NOT NULL,
    H_D_ID SMALLINT NOT NULL,
    H_W_ID INTEGER NOT NULL,
    H_DATE BIGINT NOT NULL,
    H_AMOUNT INTEGER NOT NULL,
    H_DATA CHAR(24) NOT NULL
)
    IN ts_history_025
    INDEX IN ts_history_025;
ALTER TABLE HISTORY25 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY26;
CREATE TABLE HISTORY26
(
    H_C_ID INTEGER NOT NULL,
    H_C_D_ID SMALLINT NOT NULL,
    H_C_W_ID INTEGER NOT NULL,
    H_D_ID SMALLINT NOT NULL,
    H_W_ID INTEGER NOT NULL,
    H_DATE BIGINT NOT NULL,
    H_AMOUNT INTEGER NOT NULL,
    H_DATA CHAR(24) NOT NULL
)
    IN ts_history_026
    INDEX IN ts_history_026;

```

```

ALTER TABLE HISTORY26 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY27;
CREATE TABLE HISTORY27
(
    H_C_ID INTEGER NOT NULL,
    H_C_D_ID SMALLINT NOT NULL,
    H_C_W_ID INTEGER NOT NULL,
    H_D_ID SMALLINT NOT NULL,
    H_W_ID INTEGER NOT NULL,
    H_DATE BIGINT NOT NULL,
    H_AMOUNT INTEGER NOT NULL,
    H_DATA CHAR(24) NOT NULL
)
    IN ts_history_027
    INDEX IN ts_history_027;
ALTER TABLE HISTORY27 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY28;
CREATE TABLE HISTORY28
(
    H_C_ID INTEGER NOT NULL,
    H_C_D_ID SMALLINT NOT NULL,
    H_C_W_ID INTEGER NOT NULL,
    H_D_ID SMALLINT NOT NULL,
    H_W_ID INTEGER NOT NULL,
    H_DATE BIGINT NOT NULL,
    H_AMOUNT INTEGER NOT NULL,
    H_DATA CHAR(24) NOT NULL
)
    IN ts_history_028
    INDEX IN ts_history_028;
ALTER TABLE HISTORY28 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY29;
CREATE TABLE HISTORY29
(
    H_C_ID INTEGER NOT NULL,
    H_C_D_ID SMALLINT NOT NULL,
    H_C_W_ID INTEGER NOT NULL,
    H_D_ID SMALLINT NOT NULL,
    H_W_ID INTEGER NOT NULL,
    H_DATE BIGINT NOT NULL,
    H_AMOUNT INTEGER NOT NULL,
    H_DATA CHAR(24) NOT NULL
)
    IN ts_history_029
    INDEX IN ts_history_029;
ALTER TABLE HISTORY29 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY30;
CREATE TABLE HISTORY30
(
    H_C_ID INTEGER NOT NULL,
    H_C_D_ID SMALLINT NOT NULL,
    H_C_W_ID INTEGER NOT NULL,
    H_D_ID SMALLINT NOT NULL,
    H_W_ID INTEGER NOT NULL,

```

```

    H_DATE      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
  )
  IN ts_history_030
  INDEX IN ts_history_030;
ALTER TABLE HISTORY30 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY31;
CREATE TABLE HISTORY31
  (
    H_C_ID      INTEGER NOT NULL,
    H_C_D_ID    SMALLINT NOT NULL,
    H_C_W_ID    INTEGER NOT NULL,
    H_D_ID      SMALLINT NOT NULL,
    H_W_ID      INTEGER NOT NULL,
    H_DATE      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
  )
  IN ts_history_031
  INDEX IN ts_history_031;
ALTER TABLE HISTORY31 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY32;
CREATE TABLE HISTORY32
  (
    H_C_ID      INTEGER NOT NULL,
    H_C_D_ID    SMALLINT NOT NULL,
    H_C_W_ID    INTEGER NOT NULL,
    H_D_ID      SMALLINT NOT NULL,
    H_W_ID      INTEGER NOT NULL,
    H_DATE      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
  )
  IN ts_history_032
  INDEX IN ts_history_032;
ALTER TABLE HISTORY32 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY33;
CREATE TABLE HISTORY33
  (
    H_C_ID      INTEGER NOT NULL,
    H_C_D_ID    SMALLINT NOT NULL,
    H_C_W_ID    INTEGER NOT NULL,
    H_D_ID      SMALLINT NOT NULL,
    H_W_ID      INTEGER NOT NULL,
    H_DATE      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
  )
  IN ts_history_033
  INDEX IN ts_history_033;
ALTER TABLE HISTORY33 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY34;
CREATE TABLE HISTORY34

```

```

  (
    H_C_ID      INTEGER NOT NULL,
    H_C_D_ID    SMALLINT NOT NULL,
    H_C_W_ID    INTEGER NOT NULL,
    H_D_ID      SMALLINT NOT NULL,
    H_W_ID      INTEGER NOT NULL,
    H_DATE      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
  )
  IN ts_history_034
  INDEX IN ts_history_034;
ALTER TABLE HISTORY34 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY35;
CREATE TABLE HISTORY35
  (
    H_C_ID      INTEGER NOT NULL,
    H_C_D_ID    SMALLINT NOT NULL,
    H_C_W_ID    INTEGER NOT NULL,
    H_D_ID      SMALLINT NOT NULL,
    H_W_ID      INTEGER NOT NULL,
    H_DATE      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
  )
  IN ts_history_035
  INDEX IN ts_history_035;
ALTER TABLE HISTORY35 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY36;
CREATE TABLE HISTORY36
  (
    H_C_ID      INTEGER NOT NULL,
    H_C_D_ID    SMALLINT NOT NULL,
    H_C_W_ID    INTEGER NOT NULL,
    H_D_ID      SMALLINT NOT NULL,
    H_W_ID      INTEGER NOT NULL,
    H_DATE      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
  )
  IN ts_history_036
  INDEX IN ts_history_036;
ALTER TABLE HISTORY36 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY37;
CREATE TABLE HISTORY37
  (
    H_C_ID      INTEGER NOT NULL,
    H_C_D_ID    SMALLINT NOT NULL,
    H_C_W_ID    INTEGER NOT NULL,
    H_D_ID      SMALLINT NOT NULL,
    H_W_ID      INTEGER NOT NULL,
    H_DATE      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
  )
  IN ts_history_037

```

```

  INDEX IN ts_history_037;
ALTER TABLE HISTORY37 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY38;
CREATE TABLE HISTORY38
  (
    H_C_ID      INTEGER NOT NULL,
    H_C_D_ID    SMALLINT NOT NULL,
    H_C_W_ID    INTEGER NOT NULL,
    H_D_ID      SMALLINT NOT NULL,
    H_W_ID      INTEGER NOT NULL,
    H_DATE      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
  )
  IN ts_history_038
  INDEX IN ts_history_038;
ALTER TABLE HISTORY38 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY39;
CREATE TABLE HISTORY39
  (
    H_C_ID      INTEGER NOT NULL,
    H_C_D_ID    SMALLINT NOT NULL,
    H_C_W_ID    INTEGER NOT NULL,
    H_D_ID      SMALLINT NOT NULL,
    H_W_ID      INTEGER NOT NULL,
    H_DATE      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
  )
  IN ts_history_039
  INDEX IN ts_history_039;
ALTER TABLE HISTORY39 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY40;
CREATE TABLE HISTORY40
  (
    H_C_ID      INTEGER NOT NULL,
    H_C_D_ID    SMALLINT NOT NULL,
    H_C_W_ID    INTEGER NOT NULL,
    H_D_ID      SMALLINT NOT NULL,
    H_W_ID      INTEGER NOT NULL,
    H_DATE      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
  )
  IN ts_history_040
  INDEX IN ts_history_040;
ALTER TABLE HISTORY40 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY41;
CREATE TABLE HISTORY41
  (
    H_C_ID      INTEGER NOT NULL,
    H_C_D_ID    SMALLINT NOT NULL,
    H_C_W_ID    INTEGER NOT NULL,
    H_D_ID      SMALLINT NOT NULL,

```



```

        H_W_ID    INTEGER NOT NULL,
        H_DATE    BIGINT  NOT NULL,
        H_AMOUNT  INTEGER NOT NULL,
        H_DATA    CHAR(24) NOT NULL
    )
    IN ts_history_041
    INDEX IN ts_history_041;
ALTER TABLE HISTORY41 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY42;
CREATE TABLE HISTORY42
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT  NOT NULL,
    H_AMOUNT  INTEGER NOT NULL,
    H_DATA    CHAR(24) NOT NULL
)
    IN ts_history_042
    INDEX IN ts_history_042;
ALTER TABLE HISTORY42 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY43;
CREATE TABLE HISTORY43
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT  NOT NULL,
    H_AMOUNT  INTEGER NOT NULL,
    H_DATA    CHAR(24) NOT NULL
)
    IN ts_history_043
    INDEX IN ts_history_043;
ALTER TABLE HISTORY43 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY44;
CREATE TABLE HISTORY44
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT  NOT NULL,
    H_AMOUNT  INTEGER NOT NULL,
    H_DATA    CHAR(24) NOT NULL
)
    IN ts_history_044
    INDEX IN ts_history_044;
ALTER TABLE HISTORY44 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY45;

```

```

CREATE TABLE HISTORY45
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT  NOT NULL,
    H_AMOUNT  INTEGER NOT NULL,
    H_DATA    CHAR(24) NOT NULL
)
    IN ts_history_045
    INDEX IN ts_history_045;
ALTER TABLE HISTORY45 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY46;
CREATE TABLE HISTORY46
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT  NOT NULL,
    H_AMOUNT  INTEGER NOT NULL,
    H_DATA    CHAR(24) NOT NULL
)
    IN ts_history_046
    INDEX IN ts_history_046;
ALTER TABLE HISTORY46 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY47;
CREATE TABLE HISTORY47
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT  NOT NULL,
    H_AMOUNT  INTEGER NOT NULL,
    H_DATA    CHAR(24) NOT NULL
)
    IN ts_history_047
    INDEX IN ts_history_047;
ALTER TABLE HISTORY47 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY48;
CREATE TABLE HISTORY48
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT  NOT NULL,
    H_AMOUNT  INTEGER NOT NULL,
    H_DATA    CHAR(24) NOT NULL
)

```

```

    IN ts_history_048
    INDEX IN ts_history_048;
ALTER TABLE HISTORY48 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY49;
CREATE TABLE HISTORY49
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT  NOT NULL,
    H_AMOUNT  INTEGER NOT NULL,
    H_DATA    CHAR(24) NOT NULL
)
    IN ts_history_049
    INDEX IN ts_history_049;
ALTER TABLE HISTORY49 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY50;
CREATE TABLE HISTORY50
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT  NOT NULL,
    H_AMOUNT  INTEGER NOT NULL,
    H_DATA    CHAR(24) NOT NULL
)
    IN ts_history_050
    INDEX IN ts_history_050;
ALTER TABLE HISTORY50 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY51;
CREATE TABLE HISTORY51
(
    H_C_ID    INTEGER NOT NULL,
    H_C_D_ID  SMALLINT NOT NULL,
    H_C_W_ID  INTEGER NOT NULL,
    H_D_ID    SMALLINT NOT NULL,
    H_W_ID    INTEGER NOT NULL,
    H_DATE    BIGINT  NOT NULL,
    H_AMOUNT  INTEGER NOT NULL,
    H_DATA    CHAR(24) NOT NULL
)
    IN ts_history_051
    INDEX IN ts_history_051;
ALTER TABLE HISTORY51 APPEND ON;
connect reset;

DDL/CRTB ITEM.ddl

connect to TPCC in share mode;
DROP TABLE ITEM;
CREATE TABLE ITEM

```

```

(
  I_NAME CHAR(24) NOT NULL,
  I_PRICE INTEGER NOT NULL,
  I_DATA VARCHAR(50) NOT NULL,
  I_IM_ID INTEGER NOT NULL,
  I_ID INTEGER NOT NULL
)
IN ts_item_001
INDEX IN ts_item_001
ORGANIZE BY KEY SEQUENCE (
  I_ID STARTING FROM 1 ENDING AT 100000
)
ALLOW OVERFLOW;
ALTER TABLE ITEM LOCKSIZE TABLE;
connect reset;

```

DDL/CRTB NEW ORDERA.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA1;
CREATE TABLE NEW_ORDERA1

```

```

(
  NO_O_ID INTEGER NOT NULL,
  NO_D_ID SMALLINT NOT NULL,
  NO_W_ID INTEGER NOT NULL
)

```

```

IN ts_newordA_001
INDEX IN ts_newordA_001
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 1 ENDING AT 1600,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT

```

3675

```

)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA2;
CREATE TABLE NEW_ORDERA2

```

```

(
  NO_O_ID INTEGER NOT NULL,
  NO_D_ID SMALLINT NOT NULL,
  NO_W_ID INTEGER NOT NULL
)

```

```

IN ts_newordA_002
INDEX IN ts_newordA_002
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 1601 ENDING AT

```

3200,

```

  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT

```

3675

```

)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA3;
CREATE TABLE NEW_ORDERA3

```

```

(
  NO_O_ID INTEGER NOT NULL,
  NO_D_ID SMALLINT NOT NULL,

```

```

  NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_003
INDEX IN ts_newordA_003
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 3201 ENDING AT

```

4800,

```

  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT

```

3675

```

)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA4;
CREATE TABLE NEW_ORDERA4

```

```

(
  NO_O_ID INTEGER NOT NULL,
  NO_D_ID SMALLINT NOT NULL,
  NO_W_ID INTEGER NOT NULL
)

```

```

IN ts_newordA_004
INDEX IN ts_newordA_004
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 4801 ENDING AT

```

6400,

```

  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT

```

3675

```

)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA5;
CREATE TABLE NEW_ORDERA5

```

```

(
  NO_O_ID INTEGER NOT NULL,
  NO_D_ID SMALLINT NOT NULL,
  NO_W_ID INTEGER NOT NULL
)

```

```

IN ts_newordA_005
INDEX IN ts_newordA_005
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 6401 ENDING AT

```

8000,

```

  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT

```

3675

```

)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA6;
CREATE TABLE NEW_ORDERA6

```

```

(
  NO_O_ID INTEGER NOT NULL,
  NO_D_ID SMALLINT NOT NULL,
  NO_W_ID INTEGER NOT NULL
)

```

```

IN ts_newordA_006
INDEX IN ts_newordA_006
ORGANIZE BY KEY SEQUENCE (

```

```

  NO_W_ID STARTING FROM 8001 ENDING AT

```

9600,

```

  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT

```

3675

```

)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA7;
CREATE TABLE NEW_ORDERA7

```

```

(
  NO_O_ID INTEGER NOT NULL,
  NO_D_ID SMALLINT NOT NULL,
  NO_W_ID INTEGER NOT NULL
)

```

```

IN ts_newordA_007
INDEX IN ts_newordA_007
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 9601 ENDING AT

```

11200,

```

  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT

```

3675

```

)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA8;
CREATE TABLE NEW_ORDERA8

```

```

(
  NO_O_ID INTEGER NOT NULL,
  NO_D_ID SMALLINT NOT NULL,
  NO_W_ID INTEGER NOT NULL
)

```

```

IN ts_newordA_008
INDEX IN ts_newordA_008
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 11201 ENDING AT

```

12800,

```

  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT

```

3675

```

)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA9;
CREATE TABLE NEW_ORDERA9

```

```

(
  NO_O_ID INTEGER NOT NULL,
  NO_D_ID SMALLINT NOT NULL,
  NO_W_ID INTEGER NOT NULL
)

```

```

IN ts_newordA_009
INDEX IN ts_newordA_009
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 12801 ENDING AT

```

14400,

```

  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT

```

3675

```

)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA10;
CREATE TABLE NEW_ORDERA10
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_010
INDEX IN ts_newordA_010
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 14401 ENDING AT
16000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA11;
CREATE TABLE NEW_ORDERA11
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_011
INDEX IN ts_newordA_011
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 16001 ENDING AT
17600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA12;
CREATE TABLE NEW_ORDERA12
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_012
INDEX IN ts_newordA_012
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 17601 ENDING AT
19200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA13;

```

```

CREATE TABLE NEW_ORDERA13
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_013
INDEX IN ts_newordA_013
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 19201 ENDING AT
20800,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA14;
CREATE TABLE NEW_ORDERA14
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_014
INDEX IN ts_newordA_014
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 20801 ENDING AT
22400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA15;
CREATE TABLE NEW_ORDERA15
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_015
INDEX IN ts_newordA_015
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 22401 ENDING AT
24000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA16;
CREATE TABLE NEW_ORDERA16
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)

```

```

)
IN ts_newordA_016
INDEX IN ts_newordA_016
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 24001 ENDING AT
25600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA17;
CREATE TABLE NEW_ORDERA17
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_017
INDEX IN ts_newordA_017
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 25601 ENDING AT
27200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA18;
CREATE TABLE NEW_ORDERA18
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_018
INDEX IN ts_newordA_018
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 27201 ENDING AT
28800,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA19;
CREATE TABLE NEW_ORDERA19
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_019
INDEX IN ts_newordA_019
ORGANIZE BY KEY SEQUENCE (

```

```

NO_W_ID STARTING FROM 28801 ENDING AT
30400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA20;
CREATE TABLE NEW_ORDERA20
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_020
INDEX IN ts_newordA_020
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 30401 ENDING AT
32000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA21;
CREATE TABLE NEW_ORDERA21
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_021
INDEX IN ts_newordA_021
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 32001 ENDING AT
33600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA22;
CREATE TABLE NEW_ORDERA22
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_022
INDEX IN ts_newordA_022
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 33601 ENDING AT
35200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675

```

```

)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA23;
CREATE TABLE NEW_ORDERA23
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_023
INDEX IN ts_newordA_023
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 35201 ENDING AT
36800,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA24;
CREATE TABLE NEW_ORDERA24
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_024
INDEX IN ts_newordA_024
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 36801 ENDING AT
38400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA25;
CREATE TABLE NEW_ORDERA25
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_025
INDEX IN ts_newordA_025
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 38401 ENDING AT
40000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA26;

```

```

CREATE TABLE NEW_ORDERA26
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_026
INDEX IN ts_newordA_026
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 40001 ENDING AT
41600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA27;
CREATE TABLE NEW_ORDERA27
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_027
INDEX IN ts_newordA_027
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 41601 ENDING AT
43200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA28;
CREATE TABLE NEW_ORDERA28
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_028
INDEX IN ts_newordA_028
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 43201 ENDING AT
44800,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA29;
CREATE TABLE NEW_ORDERA29
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL

```

```

)
IN ts_newordA_029
INDEX IN ts_newordA_029
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 44801 ENDING AT
46400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA30;
CREATE TABLE NEW_ORDERA30
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_030
INDEX IN ts_newordA_030
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 46401 ENDING AT
48000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA31;
CREATE TABLE NEW_ORDERA31
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_031
INDEX IN ts_newordA_031
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 48001 ENDING AT
49600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA32;
CREATE TABLE NEW_ORDERA32
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_032
INDEX IN ts_newordA_032
ORGANIZE BY KEY SEQUENCE (

```

```

NO_W_ID STARTING FROM 49601 ENDING AT
51200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA33;
CREATE TABLE NEW_ORDERA33
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_033
INDEX IN ts_newordA_033
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 51201 ENDING AT
52800,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA34;
CREATE TABLE NEW_ORDERA34
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_034
INDEX IN ts_newordA_034
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 52801 ENDING AT
54400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA35;
CREATE TABLE NEW_ORDERA35
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_035
INDEX IN ts_newordA_035
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 54401 ENDING AT
56000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)

```

```

)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA36;
CREATE TABLE NEW_ORDERA36
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_036
INDEX IN ts_newordA_036
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 56001 ENDING AT
57600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA37;
CREATE TABLE NEW_ORDERA37
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_037
INDEX IN ts_newordA_037
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 57601 ENDING AT
59200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA38;
CREATE TABLE NEW_ORDERA38
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_038
INDEX IN ts_newordA_038
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 59201 ENDING AT
60800,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA39;

```

```

CREATE TABLE NEW_ORDERA39
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_039
INDEX IN ts_newordA_039
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 60801 ENDING AT
62400,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA40;
CREATE TABLE NEW_ORDERA40
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_040
INDEX IN ts_newordA_040
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 62401 ENDING AT
64000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA41;
CREATE TABLE NEW_ORDERA41
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_041
INDEX IN ts_newordA_041
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 64001 ENDING AT
65600,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA42;
CREATE TABLE NEW_ORDERA42
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL

```

```

)
IN ts_newordA_042
INDEX IN ts_newordA_042
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 65601 ENDING AT
67200,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA43;
CREATE TABLE NEW_ORDERA43
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_043
INDEX IN ts_newordA_043
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 67201 ENDING AT
68800,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA44;
CREATE TABLE NEW_ORDERA44
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_044
INDEX IN ts_newordA_044
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 68801 ENDING AT
70400,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA45;
CREATE TABLE NEW_ORDERA45
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_045
INDEX IN ts_newordA_045
ORGANIZE BY KEY SEQUENCE (

```

```

  NO_W_ID STARTING FROM 70401 ENDING AT
72000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA46;
CREATE TABLE NEW_ORDERA46
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_046
INDEX IN ts_newordA_046
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 72001 ENDING AT
73600,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA47;
CREATE TABLE NEW_ORDERA47
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_047
INDEX IN ts_newordA_047
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 73601 ENDING AT
75200,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA48;
CREATE TABLE NEW_ORDERA48
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_048
INDEX IN ts_newordA_048
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 75201 ENDING AT
76800,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675

```

```

)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA49;
CREATE TABLE NEW_ORDERA49
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_049
INDEX IN ts_newordA_049
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 76801 ENDING AT
78400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA50;
CREATE TABLE NEW_ORDERA50
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_050
INDEX IN ts_newordA_050
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 78401 ENDING AT
80000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA51;
CREATE TABLE NEW_ORDERA51
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_051
INDEX IN ts_newordA_051
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 80001 ENDING AT
81600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

DDL/CRTB NEW ORDERB.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB1;
CREATE TABLE NEW_ORDERB1
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_001
INDEX IN ts_newordB_001
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 1 ENDING AT 1600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB2;
CREATE TABLE NEW_ORDERB2
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_002
INDEX IN ts_newordB_002
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 1601 ENDING AT
3200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB3;
CREATE TABLE NEW_ORDERB3
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_003
INDEX IN ts_newordB_003
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 3201 ENDING AT
4800,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB4;
CREATE TABLE NEW_ORDERB4
(

```

```

NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_004
INDEX IN ts_newordB_004
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 4801 ENDING AT
6400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB5;
CREATE TABLE NEW_ORDERB5
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_005
INDEX IN ts_newordB_005
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 6401 ENDING AT
8000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB6;
CREATE TABLE NEW_ORDERB6
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_006
INDEX IN ts_newordB_006
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 8001 ENDING AT
9600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB7;
CREATE TABLE NEW_ORDERB7
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_007

```

```

INDEX IN ts_newordB_007
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 9601 ENDING AT
11200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB8;
CREATE TABLE NEW_ORDERB8
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_008
INDEX IN ts_newordB_008
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 11201 ENDING AT
12800,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB9;
CREATE TABLE NEW_ORDERB9
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_009
INDEX IN ts_newordB_009
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 12801 ENDING AT
14400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB10;
CREATE TABLE NEW_ORDERB10
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_010
INDEX IN ts_newordB_010
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 14401 ENDING AT
16000,
NO_D_ID STARTING FROM 1 ENDING AT 10,

```

```

NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB11;
CREATE TABLE NEW_ORDERB11
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_011
INDEX IN ts_newordB_011
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 16001 ENDING AT
17600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB12;
CREATE TABLE NEW_ORDERB12
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_012
INDEX IN ts_newordB_012
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 17601 ENDING AT
19200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB13;
CREATE TABLE NEW_ORDERB13
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_013
INDEX IN ts_newordB_013
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 19201 ENDING AT
20800,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB14;
CREATE TABLE NEW_ORDERB14
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_014
INDEX IN ts_newordB_014
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 20801 ENDING AT
22400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB15;
CREATE TABLE NEW_ORDERB15
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_015
INDEX IN ts_newordB_015
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 22401 ENDING AT
24000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB16;
CREATE TABLE NEW_ORDERB16
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_016
INDEX IN ts_newordB_016
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 24001 ENDING AT
25600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB17;
CREATE TABLE NEW_ORDERB17
(
NO_O_ID    INTEGER    NOT NULL,

```



```

NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_017
INDEX IN ts_newordB_017
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 25601 ENDING AT
27200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB18;
CREATE TABLE NEW_ORDERB18
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_018
INDEX IN ts_newordB_018
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 27201 ENDING AT
28800,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB19;
CREATE TABLE NEW_ORDERB19
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_019
INDEX IN ts_newordB_019
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 28801 ENDING AT
30400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB20;
CREATE TABLE NEW_ORDERB20
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_020
INDEX IN ts_newordB_020

```

```

ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 30401 ENDING AT
32000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB21;
CREATE TABLE NEW_ORDERB21
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_021
INDEX IN ts_newordB_021
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 32001 ENDING AT
33600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB22;
CREATE TABLE NEW_ORDERB22
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_022
INDEX IN ts_newordB_022
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 33601 ENDING AT
35200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB23;
CREATE TABLE NEW_ORDERB23
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_023
INDEX IN ts_newordB_023
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 35201 ENDING AT
36800,
NO_D_ID STARTING FROM 1 ENDING AT 10,

```

```

NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB24;
CREATE TABLE NEW_ORDERB24
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_024
INDEX IN ts_newordB_024
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 36801 ENDING AT
38400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB25;
CREATE TABLE NEW_ORDERB25
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_025
INDEX IN ts_newordB_025
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 38401 ENDING AT
40000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB26;
CREATE TABLE NEW_ORDERB26
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_026
INDEX IN ts_newordB_026
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 40001 ENDING AT
41600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB27;
CREATE TABLE NEW_ORDERB27
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordB_027
INDEX IN ts_newordB_027
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 41601 ENDING AT
43200,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB28;
CREATE TABLE NEW_ORDERB28
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordB_028
INDEX IN ts_newordB_028
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 43201 ENDING AT
44800,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB29;
CREATE TABLE NEW_ORDERB29
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordB_029
INDEX IN ts_newordB_029
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 44801 ENDING AT
46400,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB30;
CREATE TABLE NEW_ORDERB30
(
  NO_O_ID    INTEGER NOT NULL,

```

```

  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordB_030
INDEX IN ts_newordB_030
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 46401 ENDING AT
48000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB31;
CREATE TABLE NEW_ORDERB31
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordB_031
INDEX IN ts_newordB_031
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 48001 ENDING AT
49600,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB32;
CREATE TABLE NEW_ORDERB32
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordB_032
INDEX IN ts_newordB_032
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 49601 ENDING AT
51200,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB33;
CREATE TABLE NEW_ORDERB33
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordB_033
INDEX IN ts_newordB_033

```

```

ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 51201 ENDING AT
52800,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB34;
CREATE TABLE NEW_ORDERB34
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordB_034
INDEX IN ts_newordB_034
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 52801 ENDING AT
54400,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB35;
CREATE TABLE NEW_ORDERB35
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordB_035
INDEX IN ts_newordB_035
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 54401 ENDING AT
56000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB36;
CREATE TABLE NEW_ORDERB36
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordB_036
INDEX IN ts_newordB_036
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 56001 ENDING AT
57600,
  NO_D_ID STARTING FROM 1 ENDING AT 10,

```

```

NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB37;
CREATE TABLE NEW_ORDERB37
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_037
INDEX IN ts_newordB_037
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 57601 ENDING AT
59200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB38;
CREATE TABLE NEW_ORDERB38
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_038
INDEX IN ts_newordB_038
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 59201 ENDING AT
60800,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB39;
CREATE TABLE NEW_ORDERB39
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_039
INDEX IN ts_newordB_039
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 60801 ENDING AT
62400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB40;
CREATE TABLE NEW_ORDERB40
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_040
INDEX IN ts_newordB_040
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 62401 ENDING AT
64000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB41;
CREATE TABLE NEW_ORDERB41
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_041
INDEX IN ts_newordB_041
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 64001 ENDING AT
65600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB42;
CREATE TABLE NEW_ORDERB42
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_042
INDEX IN ts_newordB_042
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 65601 ENDING AT
67200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB43;
CREATE TABLE NEW_ORDERB43
(
NO_O_ID    INTEGER    NOT NULL,

```

```

NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_043
INDEX IN ts_newordB_043
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 67201 ENDING AT
68800,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB44;
CREATE TABLE NEW_ORDERB44
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_044
INDEX IN ts_newordB_044
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 68801 ENDING AT
70400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB45;
CREATE TABLE NEW_ORDERB45
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_045
INDEX IN ts_newordB_045
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 70401 ENDING AT
72000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB46;
CREATE TABLE NEW_ORDERB46
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_046
INDEX IN ts_newordB_046

```

```

ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 72001 ENDING AT
73600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB47;
CREATE TABLE NEW_ORDERB47
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_047
INDEX IN ts_newordB_047
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 73601 ENDING AT
75200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB48;
CREATE TABLE NEW_ORDERB48
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_048
INDEX IN ts_newordB_048
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 75201 ENDING AT
76800,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB49;
CREATE TABLE NEW_ORDERB49
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_049
INDEX IN ts_newordB_049
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 76801 ENDING AT
78400,
NO_D_ID STARTING FROM 1 ENDING AT 10,

```

```

NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB50;
CREATE TABLE NEW_ORDERB50
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_050
INDEX IN ts_newordB_050
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 78401 ENDING AT
80000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB51;
CREATE TABLE NEW_ORDERB51
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_051
INDEX IN ts_newordB_051
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 80001 ENDING AT
81600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

DDL/CRTB ORDERS.ddl
connect to TPCC in share mode;
DROP TABLE ORDERS1;
CREATE TABLE ORDERS1
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT     NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT   NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT   NOT NULL
)
IN ts_order_001
INDEX IN is_order_001

```

```

ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 1 ENDING AT 1600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS2;
CREATE TABLE ORDERS2
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT     NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT   NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT   NOT NULL
)
IN ts_order_002
INDEX IN is_order_002
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 1601 ENDING AT
3200,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS3;
CREATE TABLE ORDERS3
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT     NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT   NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT   NOT NULL
)
IN ts_order_003
INDEX IN is_order_003
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 3201 ENDING AT
4800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS4;
CREATE TABLE ORDERS4
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT     NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT   NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,

```

```

O_ID      INTEGER  NOT NULL,
O_W_ID    INTEGER  NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_004
INDEX IN is_order_004
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 4801 ENDING AT
6400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS5;
CREATE TABLE ORDERS5
(
O_C_ID    INTEGER  NOT NULL,
O_ENTRY_D BIGINT   NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER  NOT NULL,
O_W_ID    INTEGER  NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_005
INDEX IN is_order_005
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 6401 ENDING AT
8000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS6;
CREATE TABLE ORDERS6
(
O_C_ID    INTEGER  NOT NULL,
O_ENTRY_D BIGINT   NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER  NOT NULL,
O_W_ID    INTEGER  NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_006
INDEX IN is_order_006
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 8001 ENDING AT
9600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS7;

```

```

CREATE TABLE ORDERS7
(
O_C_ID    INTEGER  NOT NULL,
O_ENTRY_D BIGINT   NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER  NOT NULL,
O_W_ID    INTEGER  NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_007
INDEX IN is_order_007
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 9601 ENDING AT
11200,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS8;
CREATE TABLE ORDERS8
(
O_C_ID    INTEGER  NOT NULL,
O_ENTRY_D BIGINT   NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER  NOT NULL,
O_W_ID    INTEGER  NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_008
INDEX IN is_order_008
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 11201 ENDING AT
12800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS9;
CREATE TABLE ORDERS9
(
O_C_ID    INTEGER  NOT NULL,
O_ENTRY_D BIGINT   NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER  NOT NULL,
O_W_ID    INTEGER  NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_009
INDEX IN is_order_009
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,

```

```

O_W_ID STARTING FROM 12801 ENDING AT
14400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS10;
CREATE TABLE ORDERS10
(
O_C_ID    INTEGER  NOT NULL,
O_ENTRY_D BIGINT   NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER  NOT NULL,
O_W_ID    INTEGER  NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_010
INDEX IN is_order_010
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 14401 ENDING AT
16000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS11;
CREATE TABLE ORDERS11
(
O_C_ID    INTEGER  NOT NULL,
O_ENTRY_D BIGINT   NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER  NOT NULL,
O_W_ID    INTEGER  NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_011
INDEX IN is_order_011
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 16001 ENDING AT
17600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS12;
CREATE TABLE ORDERS12
(
O_C_ID    INTEGER  NOT NULL,
O_ENTRY_D BIGINT   NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER  NOT NULL,

```

```

O_W_ID    INTEGER NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_012
INDEX IN is_order_012
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 17601 ENDING AT
19200,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS13;
CREATE TABLE ORDERS13
(
O_C_ID    INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER NOT NULL,
O_W_ID    INTEGER NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_013
INDEX IN is_order_013
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 19201 ENDING AT
20800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS14;
CREATE TABLE ORDERS14
(
O_C_ID    INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER NOT NULL,
O_W_ID    INTEGER NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_014
INDEX IN is_order_014
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 20801 ENDING AT
22400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS15;
CREATE TABLE ORDERS15

```

```

(
O_C_ID    INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER NOT NULL,
O_W_ID    INTEGER NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_015
INDEX IN is_order_015
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 22401 ENDING AT
24000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS16;
CREATE TABLE ORDERS16
(
O_C_ID    INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER NOT NULL,
O_W_ID    INTEGER NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_016
INDEX IN is_order_016
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 24001 ENDING AT
25600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS17;
CREATE TABLE ORDERS17
(
O_C_ID    INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER NOT NULL,
O_W_ID    INTEGER NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_017
INDEX IN is_order_017
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 25601 ENDING AT
27200,

```

```

O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS18;
CREATE TABLE ORDERS18
(
O_C_ID    INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER NOT NULL,
O_W_ID    INTEGER NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_018
INDEX IN is_order_018
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 27201 ENDING AT
28800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS19;
CREATE TABLE ORDERS19
(
O_C_ID    INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER NOT NULL,
O_W_ID    INTEGER NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_019
INDEX IN is_order_019
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 28801 ENDING AT
30400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS20;
CREATE TABLE ORDERS20
(
O_C_ID    INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER NOT NULL,
O_W_ID    INTEGER NOT NULL,
O_D_ID    SMALLINT NOT NULL
)

```

```

)
IN ts_order_020
INDEX IN is_order_020
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 30401 ENDING AT
32000,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS21;
CREATE TABLE ORDERS21
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_021
INDEX IN is_order_021
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 32001 ENDING AT
33600,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS22;
CREATE TABLE ORDERS22
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_022
INDEX IN is_order_022
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 33601 ENDING AT
35200,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS23;
CREATE TABLE ORDERS23
(
  O_C_ID    INTEGER NOT NULL,

```

```

  O_ENTRY_D BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_023
INDEX IN is_order_023
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 35201 ENDING AT
36800,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS24;
CREATE TABLE ORDERS24
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_024
INDEX IN is_order_024
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 36801 ENDING AT
38400,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS25;
CREATE TABLE ORDERS25
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_025
INDEX IN is_order_025
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 38401 ENDING AT
40000,
  O_D_ID STARTING FROM 1 ENDING AT 10
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS26;
CREATE TABLE ORDERS26
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_026
INDEX IN is_order_026
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 40001 ENDING AT
41600,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS27;
CREATE TABLE ORDERS27
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_027
INDEX IN is_order_027
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 41601 ENDING AT
43200,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS28;
CREATE TABLE ORDERS28
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_028

```

```

INDEX IN is_order_028
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 43201 ENDING AT
44800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS29;
CREATE TABLE ORDERS29
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_029
INDEX IN is_order_029
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 44801 ENDING AT
46400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS30;
CREATE TABLE ORDERS30
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_030
INDEX IN is_order_030
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 46401 ENDING AT
48000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS31;
CREATE TABLE ORDERS31
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,

```

```

O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_031
INDEX IN is_order_031
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 48001 ENDING AT
49600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS32;
CREATE TABLE ORDERS32
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_032
INDEX IN is_order_032
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 49601 ENDING AT
51200,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS33;
CREATE TABLE ORDERS33
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_033
INDEX IN is_order_033
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 51201 ENDING AT
52800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE ORDERS34;
CREATE TABLE ORDERS34
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_034
INDEX IN is_order_034
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 52801 ENDING AT
54400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS35;
CREATE TABLE ORDERS35
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_035
INDEX IN is_order_035
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 54401 ENDING AT
56000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS36;
CREATE TABLE ORDERS36
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_036
INDEX IN is_order_036
ORGANIZE BY KEY SEQUENCE (

```



```

O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 56001 ENDING AT
57600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS37;
CREATE TABLE ORDERS37
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_037
INDEX IN is_order_037
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 57601 ENDING AT
59200,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS38;
CREATE TABLE ORDERS38
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_038
INDEX IN is_order_038
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 59201 ENDING AT
60800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS39;
CREATE TABLE ORDERS39
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,

```

```

O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_039
INDEX IN is_order_039
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 60801 ENDING AT
62400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS40;
CREATE TABLE ORDERS40
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_040
INDEX IN is_order_040
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 62401 ENDING AT
64000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS41;
CREATE TABLE ORDERS41
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_041
INDEX IN is_order_041
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 64001 ENDING AT
65600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS42;

```

```

CREATE TABLE ORDERS42
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_042
INDEX IN is_order_042
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 65601 ENDING AT
67200,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS43;
CREATE TABLE ORDERS43
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_043
INDEX IN is_order_043
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 67201 ENDING AT
68800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS44;
CREATE TABLE ORDERS44
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_044
INDEX IN is_order_044
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,

```

```

70400, O_W_ID STARTING FROM 68801 ENDING AT
)
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS45;
CREATE TABLE ORDERS45
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_045
INDEX IN is_order_045
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 70401 ENDING AT
72000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS46;
CREATE TABLE ORDERS46
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_046
INDEX IN is_order_046
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 72001 ENDING AT
73600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS47;
CREATE TABLE ORDERS47
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,

```

```

O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_047
INDEX IN is_order_047
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 73601 ENDING AT
75200,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS48;
CREATE TABLE ORDERS48
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_048
INDEX IN is_order_048
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 75201 ENDING AT
76800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS49;
CREATE TABLE ORDERS49
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_049
INDEX IN is_order_049
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 76801 ENDING AT
78400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS50;
CREATE TABLE ORDERS50

```

```

(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_050
INDEX IN is_order_050
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 78401 ENDING AT
80000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS51;
CREATE TABLE ORDERS51
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_051
INDEX IN is_order_051
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 80001 ENDING AT
81600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

DDL/CRTB ORDER LINE.ddl
connect to TPCC in share mode;
DROP TABLE ORDER_LINE1;
CREATE TABLE ORDER_LINE1
(
OL_DELIVERY_D BIGINT NOT NULL,
OL_AMOUNT INTEGER NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)

```

```

IN ts_orderline_001
INDEX IN ts_orderline_001
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 1 ENDING AT 1600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE2;
CREATE TABLE ORDER_LINE2
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_002
INDEX IN ts_orderline_002
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 1601 ENDING AT
3200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE3;
CREATE TABLE ORDER_LINE3
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_003
INDEX IN ts_orderline_003
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 3201 ENDING AT
4800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE4;
CREATE TABLE ORDER_LINE4
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_004
INDEX IN ts_orderline_004
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 4801 ENDING AT
6400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE5;
CREATE TABLE ORDER_LINE5
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_005
INDEX IN ts_orderline_005
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 6401 ENDING AT
8000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE6;
CREATE TABLE ORDER_LINE6
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,

```

```

  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_006
INDEX IN ts_orderline_006
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 8001 ENDING AT
9600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE7;
CREATE TABLE ORDER_LINE7
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_007
INDEX IN ts_orderline_007
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 9601 ENDING AT
11200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE8;
CREATE TABLE ORDER_LINE8
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_008
INDEX IN ts_orderline_008
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 11201 ENDING AT
12800,

```

```

OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3675,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE9;
CREATE TABLE ORDER_LINE9
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_009
INDEX IN ts_orderline_009
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 12801 ENDING AT
14400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE10;
CREATE TABLE ORDER_LINE10
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_010
INDEX IN ts_orderline_010
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 14401 ENDING AT
16000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE11;
CREATE TABLE ORDER_LINE11
(

```

```

  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_011
INDEX IN ts_orderline_011
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 16001 ENDING AT
17600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE12;
CREATE TABLE ORDER_LINE12
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_012
INDEX IN ts_orderline_012
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 17601 ENDING AT
19200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE13;
CREATE TABLE ORDER_LINE13
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)

```

```

)
IN ts_orderline_013
INDEX IN ts_orderline_013
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 19201 ENDING AT
20800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE14;
CREATE TABLE ORDER_LINE14
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_014
INDEX IN ts_orderline_014
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 20801 ENDING AT
22400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE15;
CREATE TABLE ORDER_LINE15
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_015
INDEX IN ts_orderline_015
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 22401 ENDING AT
24000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE16;
CREATE TABLE ORDER_LINE16
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_016
INDEX IN ts_orderline_016
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 24001 ENDING AT
25600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE17;
CREATE TABLE ORDER_LINE17
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_017
INDEX IN ts_orderline_017
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 25601 ENDING AT
27200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE18;
CREATE TABLE ORDER_LINE18
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,

```

```

  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_018
INDEX IN ts_orderline_018
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 27201 ENDING AT
28800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE19;
CREATE TABLE ORDER_LINE19
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_019
INDEX IN ts_orderline_019
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 28801 ENDING AT
30400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE20;
CREATE TABLE ORDER_LINE20
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_020
INDEX IN ts_orderline_020
ORGANIZE BY KEY SEQUENCE (

```

```

  OL_W_ID STARTING FROM 30401 ENDING AT
32000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE21;
CREATE TABLE ORDER_LINE21
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_021
INDEX IN ts_orderline_021
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 32001 ENDING AT
33600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE22;
CREATE TABLE ORDER_LINE22
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_022
INDEX IN ts_orderline_022
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 33601 ENDING AT
35200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE23;

```

<pre> CREATE TABLE ORDER_LINE23 (OL_DELIVERY_D BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_ID INTEGER NOT NULL, OL_D_ID SMALLINT NOT NULL, OL_W_ID INTEGER NOT NULL, OL_NUMBER SMALLINT NOT NULL) IN ts_orderline_023 INDEX IN ts_orderline_023 ORGANIZE BY KEY SEQUENCE (OL_W_ID STARTING FROM 35201 ENDING AT 36800, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3675, OL_NUMBER STARTING FROM 1 ENDING AT 15) ALLOW OVERFLOW; connect reset; connect to TPCC in share mode; DROP TABLE ORDER_LINE24; CREATE TABLE ORDER_LINE24 (OL_DELIVERY_D BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_ID INTEGER NOT NULL, OL_D_ID SMALLINT NOT NULL, OL_W_ID INTEGER NOT NULL, OL_NUMBER SMALLINT NOT NULL) IN ts_orderline_024 INDEX IN ts_orderline_024 ORGANIZE BY KEY SEQUENCE (OL_W_ID STARTING FROM 36801 ENDING AT 38400, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3675, OL_NUMBER STARTING FROM 1 ENDING AT 15) ALLOW OVERFLOW; connect reset; connect to TPCC in share mode; DROP TABLE ORDER_LINE25; CREATE TABLE ORDER_LINE25 (OL_DELIVERY_D BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_ID INTEGER NOT NULL, OL_D_ID SMALLINT NOT NULL, </pre>	<pre> OL_W_ID INTEGER NOT NULL, OL_NUMBER SMALLINT NOT NULL) IN ts_orderline_025 INDEX IN ts_orderline_025 ORGANIZE BY KEY SEQUENCE (OL_W_ID STARTING FROM 38401 ENDING AT 40000, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3675, OL_NUMBER STARTING FROM 1 ENDING AT 15) ALLOW OVERFLOW; connect reset; connect to TPCC in share mode; DROP TABLE ORDER_LINE26; CREATE TABLE ORDER_LINE26 (OL_DELIVERY_D BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_ID INTEGER NOT NULL, OL_D_ID SMALLINT NOT NULL, OL_W_ID INTEGER NOT NULL, OL_NUMBER SMALLINT NOT NULL) IN ts_orderline_026 INDEX IN ts_orderline_026 ORGANIZE BY KEY SEQUENCE (OL_W_ID STARTING FROM 40001 ENDING AT 41600, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3675, OL_NUMBER STARTING FROM 1 ENDING AT 15) ALLOW OVERFLOW; connect reset; connect to TPCC in share mode; DROP TABLE ORDER_LINE27; CREATE TABLE ORDER_LINE27 (OL_DELIVERY_D BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_ID INTEGER NOT NULL, OL_D_ID SMALLINT NOT NULL, OL_W_ID INTEGER NOT NULL, OL_NUMBER SMALLINT NOT NULL) IN ts_orderline_027 INDEX IN ts_orderline_027 ORGANIZE BY KEY SEQUENCE (OL_W_ID STARTING FROM 41601 ENDING AT 43200, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3675, </pre>	<pre> OL_NUMBER STARTING FROM 1 ENDING AT 15) ALLOW OVERFLOW; connect reset; connect to TPCC in share mode; DROP TABLE ORDER_LINE28; CREATE TABLE ORDER_LINE28 (OL_DELIVERY_D BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_ID INTEGER NOT NULL, OL_D_ID SMALLINT NOT NULL, OL_W_ID INTEGER NOT NULL, OL_NUMBER SMALLINT NOT NULL) IN ts_orderline_028 INDEX IN ts_orderline_028 ORGANIZE BY KEY SEQUENCE (OL_W_ID STARTING FROM 43201 ENDING AT 44800, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3675, OL_NUMBER STARTING FROM 1 ENDING AT 15) ALLOW OVERFLOW; connect reset; connect to TPCC in share mode; DROP TABLE ORDER_LINE29; CREATE TABLE ORDER_LINE29 (OL_DELIVERY_D BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_ID INTEGER NOT NULL, OL_D_ID SMALLINT NOT NULL, OL_W_ID INTEGER NOT NULL, OL_NUMBER SMALLINT NOT NULL) IN ts_orderline_029 INDEX IN ts_orderline_029 ORGANIZE BY KEY SEQUENCE (OL_W_ID STARTING FROM 44801 ENDING AT 46400, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3675, OL_NUMBER STARTING FROM 1 ENDING AT 15) ALLOW OVERFLOW; connect reset; connect to TPCC in share mode; DROP TABLE ORDER_LINE30; CREATE TABLE ORDER_LINE30 (OL_DELIVERY_D BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, </pre>
--	--	---

```

OL_I_ID    INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID    INTEGER NOT NULL,
OL_D_ID    SMALLINT NOT NULL,
OL_W_ID    INTEGER NOT NULL,
OL_NUMBER  SMALLINT NOT NULL
)
IN ts_orderline_030
INDEX IN ts_orderline_030
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 46401 ENDING AT
48000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE31;
CREATE TABLE ORDER_LINE31
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID      INTEGER NOT NULL,
  OL_D_ID      SMALLINT NOT NULL,
  OL_W_ID      INTEGER NOT NULL,
  OL_NUMBER    SMALLINT NOT NULL
)
IN ts_orderline_031
INDEX IN ts_orderline_031
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 48001 ENDING AT
49600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE32;
CREATE TABLE ORDER_LINE32
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID      INTEGER NOT NULL,
  OL_D_ID      SMALLINT NOT NULL,
  OL_W_ID      INTEGER NOT NULL,
  OL_NUMBER    SMALLINT NOT NULL
)
IN ts_orderline_032

```

```

INDEX IN ts_orderline_032
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 49601 ENDING AT
51200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE33;
CREATE TABLE ORDER_LINE33
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID      INTEGER NOT NULL,
  OL_D_ID      SMALLINT NOT NULL,
  OL_W_ID      INTEGER NOT NULL,
  OL_NUMBER    SMALLINT NOT NULL
)
IN ts_orderline_033
INDEX IN ts_orderline_033
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 51201 ENDING AT
52800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE34;
CREATE TABLE ORDER_LINE34
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID      INTEGER NOT NULL,
  OL_D_ID      SMALLINT NOT NULL,
  OL_W_ID      INTEGER NOT NULL,
  OL_NUMBER    SMALLINT NOT NULL
)
IN ts_orderline_034
INDEX IN ts_orderline_034
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 52801 ENDING AT
54400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE35;
CREATE TABLE ORDER_LINE35
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID      INTEGER NOT NULL,
  OL_D_ID      SMALLINT NOT NULL,
  OL_W_ID      INTEGER NOT NULL,
  OL_NUMBER    SMALLINT NOT NULL
)
IN ts_orderline_035
INDEX IN ts_orderline_035
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 54401 ENDING AT
56000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE36;
CREATE TABLE ORDER_LINE36
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID      INTEGER NOT NULL,
  OL_D_ID      SMALLINT NOT NULL,
  OL_W_ID      INTEGER NOT NULL,
  OL_NUMBER    SMALLINT NOT NULL
)
IN ts_orderline_036
INDEX IN ts_orderline_036
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 56001 ENDING AT
57600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE37;
CREATE TABLE ORDER_LINE37
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,

```

```

OL_O_ID    INTEGER NOT NULL,
OL_D_ID    SMALLINT NOT NULL,
OL_W_ID    INTEGER NOT NULL,
OL_NUMBER  SMALLINT NOT NULL
)
IN ts_orderline_037
INDEX IN ts_orderline_037
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 57601 ENDING AT
59200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE38;
CREATE TABLE ORDER_LINE38
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID       INTEGER NOT NULL,
  OL_D_ID       SMALLINT NOT NULL,
  OL_W_ID       INTEGER NOT NULL,
  OL_NUMBER     SMALLINT NOT NULL
)
IN ts_orderline_038
INDEX IN ts_orderline_038
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 59201 ENDING AT
60800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE39;
CREATE TABLE ORDER_LINE39
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID       INTEGER NOT NULL,
  OL_D_ID       SMALLINT NOT NULL,
  OL_W_ID       INTEGER NOT NULL,
  OL_NUMBER     SMALLINT NOT NULL
)
IN ts_orderline_039
INDEX IN ts_orderline_039
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 60801 ENDING AT
62400,

```

```

  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE40;
CREATE TABLE ORDER_LINE40
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID       INTEGER NOT NULL,
  OL_D_ID       SMALLINT NOT NULL,
  OL_W_ID       INTEGER NOT NULL,
  OL_NUMBER     SMALLINT NOT NULL
)
IN ts_orderline_040
INDEX IN ts_orderline_040
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 62401 ENDING AT
64000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE41;
CREATE TABLE ORDER_LINE41
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID       INTEGER NOT NULL,
  OL_D_ID       SMALLINT NOT NULL,
  OL_W_ID       INTEGER NOT NULL,
  OL_NUMBER     SMALLINT NOT NULL
)
IN ts_orderline_041
INDEX IN ts_orderline_041
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 64001 ENDING AT
65600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE42;
CREATE TABLE ORDER_LINE42
(

```

```

  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID       INTEGER NOT NULL,
  OL_D_ID       SMALLINT NOT NULL,
  OL_W_ID       INTEGER NOT NULL,
  OL_NUMBER     SMALLINT NOT NULL
)
IN ts_orderline_042
INDEX IN ts_orderline_042
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 65601 ENDING AT
67200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE43;
CREATE TABLE ORDER_LINE43
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID       INTEGER NOT NULL,
  OL_D_ID       SMALLINT NOT NULL,
  OL_W_ID       INTEGER NOT NULL,
  OL_NUMBER     SMALLINT NOT NULL
)
IN ts_orderline_043
INDEX IN ts_orderline_043
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 67201 ENDING AT
68800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE44;
CREATE TABLE ORDER_LINE44
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID       INTEGER NOT NULL,
  OL_D_ID       SMALLINT NOT NULL,
  OL_W_ID       INTEGER NOT NULL,
  OL_NUMBER     SMALLINT NOT NULL
)

```



```

)
IN ts_orderline_044
INDEX IN ts_orderline_044
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 68801 ENDING AT
70400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE45;
CREATE TABLE ORDER_LINE45
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_045
INDEX IN ts_orderline_045
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 70401 ENDING AT
72000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE46;
CREATE TABLE ORDER_LINE46
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_046
INDEX IN ts_orderline_046
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 72001 ENDING AT
73600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE47;
CREATE TABLE ORDER_LINE47
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_047
INDEX IN ts_orderline_047
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 73601 ENDING AT
75200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE48;
CREATE TABLE ORDER_LINE48
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_048
INDEX IN ts_orderline_048
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 75201 ENDING AT
76800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE49;
CREATE TABLE ORDER_LINE49
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,

```

```

  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_049
INDEX IN ts_orderline_049
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 76801 ENDING AT
78400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE50;
CREATE TABLE ORDER_LINE50
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_050
INDEX IN ts_orderline_050
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 78401 ENDING AT
80000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE51;
CREATE TABLE ORDER_LINE51
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_051
INDEX IN ts_orderline_051
ORGANIZE BY KEY SEQUENCE (

```

```

81600, OL_W_ID STARTING FROM 80001 ENDING AT
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3675,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;

```

DDL/CRTB STOCK.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK1;
CREATE TABLE STOCK1
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_001
INDEX IN ts_stock_001
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 1 ENDING AT 1600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK2;
CREATE TABLE STOCK2
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID INTEGER NOT NULL
)
IN ts_stock_002
INDEX IN ts_stock_002
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 1601 ENDING AT
3200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK3;
CREATE TABLE STOCK3
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_003
INDEX IN ts_stock_003
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 3201 ENDING AT
4800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK4;
CREATE TABLE STOCK4
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID INTEGER NOT NULL
)
IN ts_stock_004
INDEX IN ts_stock_004
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 4801 ENDING AT
6400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK5;
CREATE TABLE STOCK5
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_005
INDEX IN ts_stock_005
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 6401 ENDING AT
8000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK6;
CREATE TABLE STOCK6
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_006
INDEX IN ts_stock_006
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 8001 ENDING AT
9600
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK7;
CREATE TABLE STOCK7
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA     VARCHAR(50) NOT NULL,
S_DIST_01  CHAR(24) NOT NULL,
S_DIST_02  CHAR(24) NOT NULL,
S_DIST_03  CHAR(24) NOT NULL,
S_DIST_04  CHAR(24) NOT NULL,
S_DIST_05  CHAR(24) NOT NULL,
S_DIST_06  CHAR(24) NOT NULL,
S_DIST_07  CHAR(24) NOT NULL,
S_DIST_08  CHAR(24) NOT NULL,
S_DIST_09  CHAR(24) NOT NULL,
S_DIST_10  CHAR(24) NOT NULL,
S_I_ID     INTEGER NOT NULL,
S_W_ID     INTEGER NOT NULL
)
IN ts_stock_007
INDEX IN ts_stock_007
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 9601 ENDING AT
11200
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK8;
CREATE TABLE STOCK8
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA     VARCHAR(50) NOT NULL,
S_DIST_01  CHAR(24) NOT NULL,
S_DIST_02  CHAR(24) NOT NULL,
S_DIST_03  CHAR(24) NOT NULL,
S_DIST_04  CHAR(24) NOT NULL,
S_DIST_05  CHAR(24) NOT NULL,
S_DIST_06  CHAR(24) NOT NULL,
S_DIST_07  CHAR(24) NOT NULL,
S_DIST_08  CHAR(24) NOT NULL,
S_DIST_09  CHAR(24) NOT NULL,
S_DIST_10  CHAR(24) NOT NULL,
S_I_ID     INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_008
INDEX IN ts_stock_008
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 11201 ENDING AT
12800
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK9;
CREATE TABLE STOCK9
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA     VARCHAR(50) NOT NULL,
S_DIST_01  CHAR(24) NOT NULL,
S_DIST_02  CHAR(24) NOT NULL,
S_DIST_03  CHAR(24) NOT NULL,
S_DIST_04  CHAR(24) NOT NULL,
S_DIST_05  CHAR(24) NOT NULL,
S_DIST_06  CHAR(24) NOT NULL,
S_DIST_07  CHAR(24) NOT NULL,
S_DIST_08  CHAR(24) NOT NULL,
S_DIST_09  CHAR(24) NOT NULL,
S_DIST_10  CHAR(24) NOT NULL,
S_I_ID     INTEGER NOT NULL,
S_W_ID     INTEGER NOT NULL
)
IN ts_stock_009
INDEX IN ts_stock_009
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 12801 ENDING AT
14400
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK10;
CREATE TABLE STOCK10
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA     VARCHAR(50) NOT NULL,
S_DIST_01  CHAR(24) NOT NULL,
S_DIST_02  CHAR(24) NOT NULL,
S_DIST_03  CHAR(24) NOT NULL,
S_DIST_04  CHAR(24) NOT NULL,
S_DIST_05  CHAR(24) NOT NULL,
S_DIST_06  CHAR(24) NOT NULL,
S_DIST_07  CHAR(24) NOT NULL,
S_DIST_08  CHAR(24) NOT NULL,
S_DIST_09  CHAR(24) NOT NULL,
S_DIST_10  CHAR(24) NOT NULL,
S_I_ID     INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_010
INDEX IN ts_stock_010
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 14401 ENDING AT
16000
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK11;
CREATE TABLE STOCK11
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA     VARCHAR(50) NOT NULL,
S_DIST_01  CHAR(24) NOT NULL,
S_DIST_02  CHAR(24) NOT NULL,
S_DIST_03  CHAR(24) NOT NULL,
S_DIST_04  CHAR(24) NOT NULL,
S_DIST_05  CHAR(24) NOT NULL,
S_DIST_06  CHAR(24) NOT NULL,
S_DIST_07  CHAR(24) NOT NULL,
S_DIST_08  CHAR(24) NOT NULL,
S_DIST_09  CHAR(24) NOT NULL,
S_DIST_10  CHAR(24) NOT NULL,
S_I_ID     INTEGER NOT NULL,
S_W_ID     INTEGER NOT NULL
)
IN ts_stock_011
INDEX IN ts_stock_011
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 16001 ENDING AT
17600
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK12;
CREATE TABLE STOCK12
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA     VARCHAR(50) NOT NULL,
S_DIST_01  CHAR(24) NOT NULL,
S_DIST_02  CHAR(24) NOT NULL,
S_DIST_03  CHAR(24) NOT NULL,
S_DIST_04  CHAR(24) NOT NULL,
S_DIST_05  CHAR(24) NOT NULL,
S_DIST_06  CHAR(24) NOT NULL,
S_DIST_07  CHAR(24) NOT NULL,
S_DIST_08  CHAR(24) NOT NULL,
S_DIST_09  CHAR(24) NOT NULL,
S_DIST_10  CHAR(24) NOT NULL,
S_I_ID     INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_012
INDEX IN ts_stock_012
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 17601 ENDING AT
19200
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK13;
CREATE TABLE STOCK13
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_013
INDEX IN ts_stock_013
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 19201 ENDING AT
20800
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK14;
CREATE TABLE STOCK14
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_014
INDEX IN ts_stock_014
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 20801 ENDING AT
22400
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK15;
CREATE TABLE STOCK15
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_015
INDEX IN ts_stock_015
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 22401 ENDING AT
24000
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK16;
CREATE TABLE STOCK16
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_016
INDEX IN ts_stock_016
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 24001 ENDING AT
25600
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK17;
CREATE TABLE STOCK17
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_017
INDEX IN ts_stock_017
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 25601 ENDING AT
27200
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK18;
CREATE TABLE STOCK18
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_018
INDEX IN ts_stock_018
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 27201 ENDING AT
28800
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK19;
CREATE TABLE STOCK19
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_019
INDEX IN ts_stock_019
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 28801 ENDING AT
30400
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK20;
CREATE TABLE STOCK20
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_020
INDEX IN ts_stock_020
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 30401 ENDING AT
32000
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK21;
CREATE TABLE STOCK21
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_021
INDEX IN ts_stock_021
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 32001 ENDING AT
33600
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK22;
CREATE TABLE STOCK22
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_022
INDEX IN ts_stock_022
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 33601 ENDING AT
35200
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK23;
CREATE TABLE STOCK23
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_023
INDEX IN ts_stock_023
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 35201 ENDING AT
36800
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK24;
CREATE TABLE STOCK24
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_024
INDEX IN ts_stock_024
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 38401 ENDING AT
38400
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK25;
CREATE TABLE STOCK25
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_025
INDEX IN ts_stock_025
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 38401 ENDING AT
40000
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK26;
CREATE TABLE STOCK26
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_026
INDEX IN ts_stock_026
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 40001 ENDING AT
41600
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK27;
CREATE TABLE STOCK27
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_027
INDEX IN ts_stock_027
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 41601 ENDING AT
43200
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK28;
CREATE TABLE STOCK28
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_028
INDEX IN ts_stock_028
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 43201 ENDING AT
44800
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK29;
CREATE TABLE STOCK29
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_029
INDEX IN ts_stock_029
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 44801 ENDING AT
46400
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK30;
CREATE TABLE STOCK30
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_030
INDEX IN ts_stock_030
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 46401 ENDING AT
48000
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK31;
CREATE TABLE STOCK31
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_031
INDEX IN ts_stock_031
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 48001 ENDING AT
49600
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK32;
CREATE TABLE STOCK32
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_032
INDEX IN ts_stock_032
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 49601 ENDING AT
51200
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK33;
CREATE TABLE STOCK33
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_033
INDEX IN ts_stock_033
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 51201 ENDING AT
52800
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK34;
CREATE TABLE STOCK34
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_034
INDEX IN ts_stock_034
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 52801 ENDING AT
54400
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK35;
CREATE TABLE STOCK35
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_035
INDEX IN ts_stock_035
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 54401 ENDING AT
56000
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK36;
CREATE TABLE STOCK36
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_036
INDEX IN ts_stock_036
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 56001 ENDING AT
57600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK37;
CREATE TABLE STOCK37
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_037
INDEX IN ts_stock_037
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 57601 ENDING AT
59200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK38;
CREATE TABLE STOCK38
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_038
INDEX IN ts_stock_038
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 59201 ENDING AT
60800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK39;
CREATE TABLE STOCK39
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_039
INDEX IN ts_stock_039
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 60801 ENDING AT
62400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK40;
CREATE TABLE STOCK40
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_040
INDEX IN ts_stock_040
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 62401 ENDING AT
64000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK41;
CREATE TABLE STOCK41
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_041
INDEX IN ts_stock_041
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 64001 ENDING AT
65600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK42;
CREATE TABLE STOCK42
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```



```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_042
INDEX IN ts_stock_042
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 65601 ENDING AT
67200
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK43;
CREATE TABLE STOCK43
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_043
INDEX IN ts_stock_043
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 67201 ENDING AT
68800
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK44;
CREATE TABLE STOCK44
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_044
INDEX IN ts_stock_044
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 68801 ENDING AT
70400
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK45;
CREATE TABLE STOCK45
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_045
INDEX IN ts_stock_045
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 70401 ENDING AT
72000
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK46;
CREATE TABLE STOCK46
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_046
INDEX IN ts_stock_046
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 72001 ENDING AT
73600
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK47;
CREATE TABLE STOCK47
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_047
INDEX IN ts_stock_047
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 73601 ENDING AT
75200
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK48;
CREATE TABLE STOCK48
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_048
INDEX IN ts_stock_048
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 75201 ENDING AT
76800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK49;
CREATE TABLE STOCK49
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD      INTEGER NOT NULL,
S_DATA     VARCHAR(50) NOT NULL,
S_DIST_01  CHAR(24) NOT NULL,
S_DIST_02  CHAR(24) NOT NULL,
S_DIST_03  CHAR(24) NOT NULL,
S_DIST_04  CHAR(24) NOT NULL,
S_DIST_05  CHAR(24) NOT NULL,
S_DIST_06  CHAR(24) NOT NULL,
S_DIST_07  CHAR(24) NOT NULL,
S_DIST_08  CHAR(24) NOT NULL,
S_DIST_09  CHAR(24) NOT NULL,
S_DIST_10  CHAR(24) NOT NULL,
S_I_ID     INTEGER NOT NULL,
S_W_ID     INTEGER NOT NULL
)
IN ts_stock_049
INDEX IN ts_stock_049
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 76801 ENDING AT
78400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK50;
CREATE TABLE STOCK50
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD      INTEGER NOT NULL,
S_DATA     VARCHAR(50) NOT NULL,
S_DIST_01  CHAR(24) NOT NULL,
S_DIST_02  CHAR(24) NOT NULL,
S_DIST_03  CHAR(24) NOT NULL,
S_DIST_04  CHAR(24) NOT NULL,
S_DIST_05  CHAR(24) NOT NULL,
S_DIST_06  CHAR(24) NOT NULL,
S_DIST_07  CHAR(24) NOT NULL,
S_DIST_08  CHAR(24) NOT NULL,
S_DIST_09  CHAR(24) NOT NULL,
S_DIST_10  CHAR(24) NOT NULL,
S_I_ID     INTEGER NOT NULL,

```

```

S_W_ID    INTEGER    NOT NULL
)
IN ts_stock_050
INDEX IN ts_stock_050
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 78401 ENDING AT
80000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK51;
CREATE TABLE STOCK51
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD      INTEGER NOT NULL,
S_DATA     VARCHAR(50) NOT NULL,
S_DIST_01  CHAR(24) NOT NULL,
S_DIST_02  CHAR(24) NOT NULL,
S_DIST_03  CHAR(24) NOT NULL,
S_DIST_04  CHAR(24) NOT NULL,
S_DIST_05  CHAR(24) NOT NULL,
S_DIST_06  CHAR(24) NOT NULL,
S_DIST_07  CHAR(24) NOT NULL,
S_DIST_08  CHAR(24) NOT NULL,
S_DIST_09  CHAR(24) NOT NULL,
S_DIST_10  CHAR(24) NOT NULL,
S_I_ID     INTEGER NOT NULL,
S_W_ID     INTEGER NOT NULL
)
IN ts_stock_051
INDEX IN ts_stock_051
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 80001 ENDING AT
81600
)
ALLOW OVERFLOW;
connect reset;

DDL/CRTB_WAREHOUSE.ddl
connect to TPCC in share mode;
DROP TABLE WAREHOUSE1;
CREATE TABLE WAREHOUSE1
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_001
INDEX IN ts_ware_001

```

```

ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 1 ENDING AT 1600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE2;
CREATE TABLE WAREHOUSE2
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_002
INDEX IN ts_ware_002
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 1601 ENDING AT 3200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE3;
CREATE TABLE WAREHOUSE3
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_003
INDEX IN ts_ware_003
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 3201 ENDING AT 4800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE4;
CREATE TABLE WAREHOUSE4
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_004

```

```

INDEX IN ts_ware_004
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 4801 ENDING AT 6400
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE5;
CREATE TABLE WAREHOUSE5
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
)
IN ts_ware_005
INDEX IN ts_ware_005
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 6401 ENDING AT 8000
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE6;
CREATE TABLE WAREHOUSE6
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
)
IN ts_ware_006
INDEX IN ts_ware_006
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 8001 ENDING AT 9600
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE7;
CREATE TABLE WAREHOUSE7
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
)

```

```

IN ts_ware_007
INDEX IN ts_ware_007
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 9601 ENDING AT 11200
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE8;
CREATE TABLE WAREHOUSE8
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
)
IN ts_ware_008
INDEX IN ts_ware_008
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 11201 ENDING AT 12800
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE9;
CREATE TABLE WAREHOUSE9
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
)
IN ts_ware_009
INDEX IN ts_ware_009
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 12801 ENDING AT 14400
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE10;
CREATE TABLE WAREHOUSE10
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
)

```

```

)
)
IN ts_ware_010
INDEX IN ts_ware_010
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 14401 ENDING AT 16000
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE11;
CREATE TABLE WAREHOUSE11
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
)
IN ts_ware_011
INDEX IN ts_ware_011
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 16001 ENDING AT 17600
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE12;
CREATE TABLE WAREHOUSE12
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
)
IN ts_ware_012
INDEX IN ts_ware_012
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 17601 ENDING AT 19200
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE13;
CREATE TABLE WAREHOUSE13
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
)

```

```

        W_ID    INTEGER    NOT NULL
    )
    IN ts_ware_013
    INDEX IN ts_ware_013
    ORGANIZE BY KEY SEQUENCE (
        W_ID STARTING FROM 19201 ENDING AT 20800
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE14;
CREATE TABLE WAREHOUSE14
(
    W_NAME    CHAR(10)    NOT NULL,
    W_STREET_1 CHAR(20)    NOT NULL,
    W_STREET_2 CHAR(20)    NOT NULL,
    W_CITY    CHAR(20)    NOT NULL,
    W_STATE   CHAR(2)     NOT NULL,
    W_ZIP     CHAR(9)     NOT NULL,
    W_TAX     INTEGER    NOT NULL,
    W_YTD     BIGINT     NOT NULL,
    W_ID      INTEGER    NOT NULL
)
    IN ts_ware_014
    INDEX IN ts_ware_014
    ORGANIZE BY KEY SEQUENCE (
        W_ID STARTING FROM 20801 ENDING AT 22400
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE15;
CREATE TABLE WAREHOUSE15
(
    W_NAME    CHAR(10)    NOT NULL,
    W_STREET_1 CHAR(20)    NOT NULL,
    W_STREET_2 CHAR(20)    NOT NULL,
    W_CITY    CHAR(20)    NOT NULL,
    W_STATE   CHAR(2)     NOT NULL,
    W_ZIP     CHAR(9)     NOT NULL,
    W_TAX     INTEGER    NOT NULL,
    W_YTD     BIGINT     NOT NULL,
    W_ID      INTEGER    NOT NULL
)
    IN ts_ware_015
    INDEX IN ts_ware_015
    ORGANIZE BY KEY SEQUENCE (
        W_ID STARTING FROM 22401 ENDING AT 24000
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE16;
CREATE TABLE WAREHOUSE16
(
    W_NAME    CHAR(10)    NOT NULL,
    W_STREET_1 CHAR(20)    NOT NULL,
    W_STREET_2 CHAR(20)    NOT NULL,
    W_CITY    CHAR(20)    NOT NULL,
    W_STATE   CHAR(2)     NOT NULL,
    W_ZIP     CHAR(9)     NOT NULL,
    W_TAX     INTEGER    NOT NULL,

```

```

        W_YTD   BIGINT     NOT NULL,
        W_ID    INTEGER    NOT NULL
    )
    IN ts_ware_016
    INDEX IN ts_ware_016
    ORGANIZE BY KEY SEQUENCE (
        W_ID STARTING FROM 24001 ENDING AT 25600
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE17;
CREATE TABLE WAREHOUSE17
(
    W_NAME    CHAR(10)    NOT NULL,
    W_STREET_1 CHAR(20)    NOT NULL,
    W_STREET_2 CHAR(20)    NOT NULL,
    W_CITY    CHAR(20)    NOT NULL,
    W_STATE   CHAR(2)     NOT NULL,
    W_ZIP     CHAR(9)     NOT NULL,
    W_TAX     INTEGER    NOT NULL,
    W_YTD     BIGINT     NOT NULL,
    W_ID      INTEGER    NOT NULL
)
    IN ts_ware_017
    INDEX IN ts_ware_017
    ORGANIZE BY KEY SEQUENCE (
        W_ID STARTING FROM 25601 ENDING AT 27200
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE18;
CREATE TABLE WAREHOUSE18
(
    W_NAME    CHAR(10)    NOT NULL,
    W_STREET_1 CHAR(20)    NOT NULL,
    W_STREET_2 CHAR(20)    NOT NULL,
    W_CITY    CHAR(20)    NOT NULL,
    W_STATE   CHAR(2)     NOT NULL,
    W_ZIP     CHAR(9)     NOT NULL,
    W_TAX     INTEGER    NOT NULL,
    W_YTD     BIGINT     NOT NULL,
    W_ID      INTEGER    NOT NULL
)
    IN ts_ware_018
    INDEX IN ts_ware_018
    ORGANIZE BY KEY SEQUENCE (
        W_ID STARTING FROM 27201 ENDING AT 28800
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE19;
CREATE TABLE WAREHOUSE19
(
    W_NAME    CHAR(10)    NOT NULL,
    W_STREET_1 CHAR(20)    NOT NULL,
    W_STREET_2 CHAR(20)    NOT NULL,
    W_CITY    CHAR(20)    NOT NULL,
    W_STATE   CHAR(2)     NOT NULL,
    W_ZIP     CHAR(9)     NOT NULL,

```

```

        W_TAX   INTEGER    NOT NULL,
        W_YTD   BIGINT     NOT NULL,
        W_ID    INTEGER    NOT NULL
    )
    IN ts_ware_019
    INDEX IN ts_ware_019
    ORGANIZE BY KEY SEQUENCE (
        W_ID STARTING FROM 28801 ENDING AT 30400
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE20;
CREATE TABLE WAREHOUSE20
(
    W_NAME    CHAR(10)    NOT NULL,
    W_STREET_1 CHAR(20)    NOT NULL,
    W_STREET_2 CHAR(20)    NOT NULL,
    W_CITY    CHAR(20)    NOT NULL,
    W_STATE   CHAR(2)     NOT NULL,
    W_ZIP     CHAR(9)     NOT NULL,
    W_TAX     INTEGER    NOT NULL,
    W_YTD     BIGINT     NOT NULL,
    W_ID      INTEGER    NOT NULL
)
    IN ts_ware_020
    INDEX IN ts_ware_020
    ORGANIZE BY KEY SEQUENCE (
        W_ID STARTING FROM 30401 ENDING AT 32000
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE21;
CREATE TABLE WAREHOUSE21
(
    W_NAME    CHAR(10)    NOT NULL,
    W_STREET_1 CHAR(20)    NOT NULL,
    W_STREET_2 CHAR(20)    NOT NULL,
    W_CITY    CHAR(20)    NOT NULL,
    W_STATE   CHAR(2)     NOT NULL,
    W_ZIP     CHAR(9)     NOT NULL,
    W_TAX     INTEGER    NOT NULL,
    W_YTD     BIGINT     NOT NULL,
    W_ID      INTEGER    NOT NULL
)
    IN ts_ware_021
    INDEX IN ts_ware_021
    ORGANIZE BY KEY SEQUENCE (
        W_ID STARTING FROM 32001 ENDING AT 33600
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE22;
CREATE TABLE WAREHOUSE22
(
    W_NAME    CHAR(10)    NOT NULL,
    W_STREET_1 CHAR(20)    NOT NULL,
    W_STREET_2 CHAR(20)    NOT NULL,
    W_CITY    CHAR(20)    NOT NULL,
    W_STATE   CHAR(2)     NOT NULL,

```

```

W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_022
INDEX IN ts_ware_022
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 33601 ENDING AT 35200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE23;
CREATE TABLE WAREHOUSE23
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_023
INDEX IN ts_ware_023
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 35201 ENDING AT 36800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE24;
CREATE TABLE WAREHOUSE24
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_024
INDEX IN ts_ware_024
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 36801 ENDING AT 38400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE25;
CREATE TABLE WAREHOUSE25
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_025
INDEX IN ts_ware_025
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 38401 ENDING AT 40000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE26;
CREATE TABLE WAREHOUSE26
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_026
INDEX IN ts_ware_026
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 40001 ENDING AT 41600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE27;
CREATE TABLE WAREHOUSE27
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_027
INDEX IN ts_ware_027
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 41601 ENDING AT 43200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE28;
CREATE TABLE WAREHOUSE28
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,

```

```

W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_028
INDEX IN ts_ware_028
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 43201 ENDING AT 44800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE29;
CREATE TABLE WAREHOUSE29
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_029
INDEX IN ts_ware_029
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 44801 ENDING AT 46400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE30;
CREATE TABLE WAREHOUSE30
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_030
INDEX IN ts_ware_030
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 46401 ENDING AT 48000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE31;
CREATE TABLE WAREHOUSE31
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,

```

```

W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_031
INDEX IN ts_ware_031
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 48001 ENDING AT 49600
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE32;
CREATE TABLE WAREHOUSE32
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_032
INDEX IN ts_ware_032
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 49601 ENDING AT 51200
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE33;
CREATE TABLE WAREHOUSE33
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_033
INDEX IN ts_ware_033
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 51201 ENDING AT 52800
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE34;
CREATE TABLE WAREHOUSE34
(
W_NAME CHAR(10) NOT NULL,

```

```

W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_034
INDEX IN ts_ware_034
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 52801 ENDING AT 54400
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE35;
CREATE TABLE WAREHOUSE35
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_035
INDEX IN ts_ware_035
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 54401 ENDING AT 56000
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE36;
CREATE TABLE WAREHOUSE36
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_036
INDEX IN ts_ware_036
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 56001 ENDING AT 57600
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE37;
CREATE TABLE WAREHOUSE37
(

```

```

W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_037
INDEX IN ts_ware_037
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 57601 ENDING AT 59200
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE38;
CREATE TABLE WAREHOUSE38
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_038
INDEX IN ts_ware_038
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 59201 ENDING AT 60800
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE39;
CREATE TABLE WAREHOUSE39
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_039
INDEX IN ts_ware_039
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 60801 ENDING AT 62400
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE40;
CREATE TABLE WAREHOUSE40

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_040
INDEX IN ts_ware_040
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 62401 ENDING AT 64000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE41;
CREATE TABLE WAREHOUSE41
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_041
INDEX IN ts_ware_041
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 64001 ENDING AT 65600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE42;
CREATE TABLE WAREHOUSE42
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_042
INDEX IN ts_ware_042
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 65601 ENDING AT 67200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE43;

```

```

CREATE TABLE WAREHOUSE43
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_043
INDEX IN ts_ware_043
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 67201 ENDING AT 68800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE44;
CREATE TABLE WAREHOUSE44
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_044
INDEX IN ts_ware_044
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 68801 ENDING AT 70400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE45;
CREATE TABLE WAREHOUSE45
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_045
INDEX IN ts_ware_045
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 70401 ENDING AT 72000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;

```

```

DROP TABLE WAREHOUSE46;
CREATE TABLE WAREHOUSE46
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_046
INDEX IN ts_ware_046
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 72001 ENDING AT 73600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE47;
CREATE TABLE WAREHOUSE47
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_047
INDEX IN ts_ware_047
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 73601 ENDING AT 75200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE48;
CREATE TABLE WAREHOUSE48
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_048
INDEX IN ts_ware_048
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 75201 ENDING AT 76800
)
ALLOW OVERFLOW;
connect reset;

```

```
connect to TPCC in share mode;
DROP TABLE WAREHOUSE49;
CREATE TABLE WAREHOUSE49
```

```
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_049
INDEX IN ts_ware_049
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 76801 ENDING AT 78400
)
ALLOW OVERFLOW;
```

```
connect reset;
```

```
connect to TPCC in share mode;
DROP TABLE WAREHOUSE50;
CREATE TABLE WAREHOUSE50
```

```
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_050
INDEX IN ts_ware_050
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 78401 ENDING AT 80000
)
ALLOW OVERFLOW;
```

```
connect reset;
```

```
connect to TPCC in share mode;
DROP TABLE WAREHOUSE51;
CREATE TABLE WAREHOUSE51
```

```
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_051
INDEX IN ts_ware_051
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 80001 ENDING AT 81600
)
ALLOW OVERFLOW;
```

```
connect reset;
```

DDL/CRVW CUSTOMER.ddl

```
connect to TPCC in share mode;
DROP VIEW CUSTOMER;
CREATE VIEW CUSTOMER
```

```
(C_ID,
  C_STATE,
  C_ZIP,
  C_PHONE,
  C_SINCE,
  C_CREDIT_LIM,
  C_MIDDLE,
  C_CREDIT,
  C_DISCOUNT,
  C_DATA,
  C_LAST,
  C_FIRST,
  C_STREET_1,
  C_STREET_2,
  C_CITY,
  C_D_ID,
  C_W_ID,
  C_DELIVERY_CNT,
  C_BALANCE,
  C_YTD_PAYMENT,
  C_PAYMENT_CNT
) AS SELECT * FROM CUSTOMER1 UNION ALL
```

```
SELECT * FROM CUSTOMER2 UNION ALL
SELECT * FROM CUSTOMER3 UNION ALL
SELECT * FROM CUSTOMER4 UNION ALL
SELECT * FROM CUSTOMER5 UNION ALL
SELECT * FROM CUSTOMER6 UNION ALL
SELECT * FROM CUSTOMER7 UNION ALL
SELECT * FROM CUSTOMER8 UNION ALL
SELECT * FROM CUSTOMER9 UNION ALL
SELECT * FROM CUSTOMER10 UNION ALL
SELECT * FROM CUSTOMER11 UNION ALL
SELECT * FROM CUSTOMER12 UNION ALL
SELECT * FROM CUSTOMER13 UNION ALL
SELECT * FROM CUSTOMER14 UNION ALL
SELECT * FROM CUSTOMER15 UNION ALL
SELECT * FROM CUSTOMER16 UNION ALL
SELECT * FROM CUSTOMER17 UNION ALL
SELECT * FROM CUSTOMER18 UNION ALL
SELECT * FROM CUSTOMER19 UNION ALL
SELECT * FROM CUSTOMER20 UNION ALL
SELECT * FROM CUSTOMER21 UNION ALL
SELECT * FROM CUSTOMER22 UNION ALL
SELECT * FROM CUSTOMER23 UNION ALL
SELECT * FROM CUSTOMER24 UNION ALL
SELECT * FROM CUSTOMER25 UNION ALL
SELECT * FROM CUSTOMER26 UNION ALL
SELECT * FROM CUSTOMER27 UNION ALL
SELECT * FROM CUSTOMER28 UNION ALL
SELECT * FROM CUSTOMER29 UNION ALL
SELECT * FROM CUSTOMER30 UNION ALL
SELECT * FROM CUSTOMER31 UNION ALL
SELECT * FROM CUSTOMER32 UNION ALL
SELECT * FROM CUSTOMER33 UNION ALL
```

```
SELECT * FROM CUSTOMER34 UNION ALL
SELECT * FROM CUSTOMER35 UNION ALL
SELECT * FROM CUSTOMER36 UNION ALL
SELECT * FROM CUSTOMER37 UNION ALL
SELECT * FROM CUSTOMER38 UNION ALL
SELECT * FROM CUSTOMER39 UNION ALL
SELECT * FROM CUSTOMER40 UNION ALL
SELECT * FROM CUSTOMER41 UNION ALL
SELECT * FROM CUSTOMER42 UNION ALL
SELECT * FROM CUSTOMER43 UNION ALL
SELECT * FROM CUSTOMER44 UNION ALL
SELECT * FROM CUSTOMER45 UNION ALL
SELECT * FROM CUSTOMER46 UNION ALL
SELECT * FROM CUSTOMER47 UNION ALL
SELECT * FROM CUSTOMER48 UNION ALL
SELECT * FROM CUSTOMER49 UNION ALL
SELECT * FROM CUSTOMER50 UNION ALL
SELECT * FROM CUSTOMER51
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

DDL/CRVW DISTRICT.ddl

```
connect to TPCC in share mode;
DROP VIEW DISTRICT;
CREATE VIEW DISTRICT
```

```
(D_NEXT_O_ID,
  D_TAX,
  D_YTD,
  D_NAME,
  D_STREET_1,
  D_STREET_2,
  D_CITY,
  D_STATE,
  D_ZIP,
  D_ID,
  D_W_ID
) AS SELECT * FROM DISTRICT1 UNION ALL
```

```
SELECT * FROM DISTRICT2 UNION ALL
SELECT * FROM DISTRICT3 UNION ALL
SELECT * FROM DISTRICT4 UNION ALL
SELECT * FROM DISTRICT5 UNION ALL
SELECT * FROM DISTRICT6 UNION ALL
SELECT * FROM DISTRICT7 UNION ALL
SELECT * FROM DISTRICT8 UNION ALL
SELECT * FROM DISTRICT9 UNION ALL
SELECT * FROM DISTRICT10 UNION ALL
SELECT * FROM DISTRICT11 UNION ALL
SELECT * FROM DISTRICT12 UNION ALL
SELECT * FROM DISTRICT13 UNION ALL
SELECT * FROM DISTRICT14 UNION ALL
SELECT * FROM DISTRICT15 UNION ALL
SELECT * FROM DISTRICT16 UNION ALL
SELECT * FROM DISTRICT17 UNION ALL
SELECT * FROM DISTRICT18 UNION ALL
SELECT * FROM DISTRICT19 UNION ALL
SELECT * FROM DISTRICT20 UNION ALL
SELECT * FROM DISTRICT21 UNION ALL
SELECT * FROM DISTRICT22 UNION ALL
SELECT * FROM DISTRICT23 UNION ALL
```



```

SELECT * FROM DISTRICT24 UNION ALL
SELECT * FROM DISTRICT25 UNION ALL
SELECT * FROM DISTRICT26 UNION ALL
SELECT * FROM DISTRICT27 UNION ALL
SELECT * FROM DISTRICT28 UNION ALL
SELECT * FROM DISTRICT29 UNION ALL
SELECT * FROM DISTRICT30 UNION ALL
SELECT * FROM DISTRICT31 UNION ALL
SELECT * FROM DISTRICT32 UNION ALL
SELECT * FROM DISTRICT33 UNION ALL
SELECT * FROM DISTRICT34 UNION ALL
SELECT * FROM DISTRICT35 UNION ALL
SELECT * FROM DISTRICT36 UNION ALL
SELECT * FROM DISTRICT37 UNION ALL
SELECT * FROM DISTRICT38 UNION ALL
SELECT * FROM DISTRICT39 UNION ALL
SELECT * FROM DISTRICT40 UNION ALL
SELECT * FROM DISTRICT41 UNION ALL
SELECT * FROM DISTRICT42 UNION ALL
SELECT * FROM DISTRICT43 UNION ALL
SELECT * FROM DISTRICT44 UNION ALL
SELECT * FROM DISTRICT45 UNION ALL
SELECT * FROM DISTRICT46 UNION ALL
SELECT * FROM DISTRICT47 UNION ALL
SELECT * FROM DISTRICT48 UNION ALL
SELECT * FROM DISTRICT49 UNION ALL
SELECT * FROM DISTRICT50 UNION ALL
SELECT * FROM DISTRICT51
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

DDL/CRVW HISTORY.ddl

```

connect to TPCC in share mode;
DROP VIEW HISTORY;
CREATE VIEW HISTORY
(H_C_ID,
 H_C_D_ID,
 H_C_W_ID,
 H_D_ID,
 H_W_ID,
 H_DATE,
 H_AMOUNT,
 H_DATA
) AS SELECT * FROM HISTORY1 UNION ALL
SELECT * FROM HISTORY2 UNION ALL
SELECT * FROM HISTORY3 UNION ALL
SELECT * FROM HISTORY4 UNION ALL
SELECT * FROM HISTORY5 UNION ALL
SELECT * FROM HISTORY6 UNION ALL
SELECT * FROM HISTORY7 UNION ALL
SELECT * FROM HISTORY8 UNION ALL
SELECT * FROM HISTORY9 UNION ALL
SELECT * FROM HISTORY10 UNION ALL
SELECT * FROM HISTORY11 UNION ALL
SELECT * FROM HISTORY12 UNION ALL
SELECT * FROM HISTORY13 UNION ALL
SELECT * FROM HISTORY14 UNION ALL
SELECT * FROM HISTORY15 UNION ALL
SELECT * FROM HISTORY16 UNION ALL

```

```

SELECT * FROM HISTORY17 UNION ALL
SELECT * FROM HISTORY18 UNION ALL
SELECT * FROM HISTORY19 UNION ALL
SELECT * FROM HISTORY20 UNION ALL
SELECT * FROM HISTORY21 UNION ALL
SELECT * FROM HISTORY22 UNION ALL
SELECT * FROM HISTORY23 UNION ALL
SELECT * FROM HISTORY24 UNION ALL
SELECT * FROM HISTORY25 UNION ALL
SELECT * FROM HISTORY26 UNION ALL
SELECT * FROM HISTORY27 UNION ALL
SELECT * FROM HISTORY28 UNION ALL
SELECT * FROM HISTORY29 UNION ALL
SELECT * FROM HISTORY30 UNION ALL
SELECT * FROM HISTORY31 UNION ALL
SELECT * FROM HISTORY32 UNION ALL
SELECT * FROM HISTORY33 UNION ALL
SELECT * FROM HISTORY34 UNION ALL
SELECT * FROM HISTORY35 UNION ALL
SELECT * FROM HISTORY36 UNION ALL
SELECT * FROM HISTORY37 UNION ALL
SELECT * FROM HISTORY38 UNION ALL
SELECT * FROM HISTORY39 UNION ALL
SELECT * FROM HISTORY40 UNION ALL
SELECT * FROM HISTORY41 UNION ALL
SELECT * FROM HISTORY42 UNION ALL
SELECT * FROM HISTORY43 UNION ALL
SELECT * FROM HISTORY44 UNION ALL
SELECT * FROM HISTORY45 UNION ALL
SELECT * FROM HISTORY46 UNION ALL
SELECT * FROM HISTORY47 UNION ALL
SELECT * FROM HISTORY48 UNION ALL
SELECT * FROM HISTORY49 UNION ALL
SELECT * FROM HISTORY50 UNION ALL
SELECT * FROM HISTORY51
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

DDL/CRVW NEW ORDER.ddl

```

connect to TPCC in share mode;
DROP VIEW NEW_ORDER;
CREATE VIEW NEW_ORDER
(NO_O_ID,
 NO_D_ID,
 NO_W_ID
) AS SELECT * FROM NEW_ORDERA1 UNION
ALL
SELECT * FROM NEW_ORDERA2 UNION ALL
SELECT * FROM NEW_ORDERA3 UNION ALL
SELECT * FROM NEW_ORDERA4 UNION ALL
SELECT * FROM NEW_ORDERA5 UNION ALL
SELECT * FROM NEW_ORDERA6 UNION ALL
SELECT * FROM NEW_ORDERA7 UNION ALL
SELECT * FROM NEW_ORDERA8 UNION ALL
SELECT * FROM NEW_ORDERA9 UNION ALL
SELECT * FROM NEW_ORDERA10 UNION ALL
SELECT * FROM NEW_ORDERA11 UNION ALL
SELECT * FROM NEW_ORDERA12 UNION ALL
SELECT * FROM NEW_ORDERA13 UNION ALL

```

```

SELECT * FROM NEW_ORDERA14 UNION ALL
SELECT * FROM NEW_ORDERA15 UNION ALL
SELECT * FROM NEW_ORDERA16 UNION ALL
SELECT * FROM NEW_ORDERA17 UNION ALL
SELECT * FROM NEW_ORDERA18 UNION ALL
SELECT * FROM NEW_ORDERA19 UNION ALL
SELECT * FROM NEW_ORDERA20 UNION ALL
SELECT * FROM NEW_ORDERA21 UNION ALL
SELECT * FROM NEW_ORDERA22 UNION ALL
SELECT * FROM NEW_ORDERA23 UNION ALL
SELECT * FROM NEW_ORDERA24 UNION ALL
SELECT * FROM NEW_ORDERA25 UNION ALL
SELECT * FROM NEW_ORDERA26 UNION ALL
SELECT * FROM NEW_ORDERA27 UNION ALL
SELECT * FROM NEW_ORDERA28 UNION ALL
SELECT * FROM NEW_ORDERA29 UNION ALL
SELECT * FROM NEW_ORDERA30 UNION ALL
SELECT * FROM NEW_ORDERA31 UNION ALL
SELECT * FROM NEW_ORDERA32 UNION ALL
SELECT * FROM NEW_ORDERA33 UNION ALL
SELECT * FROM NEW_ORDERA34 UNION ALL
SELECT * FROM NEW_ORDERA35 UNION ALL
SELECT * FROM NEW_ORDERA36 UNION ALL
SELECT * FROM NEW_ORDERA37 UNION ALL
SELECT * FROM NEW_ORDERA38 UNION ALL
SELECT * FROM NEW_ORDERA39 UNION ALL
SELECT * FROM NEW_ORDERA40 UNION ALL
SELECT * FROM NEW_ORDERA41 UNION ALL
SELECT * FROM NEW_ORDERA42 UNION ALL
SELECT * FROM NEW_ORDERA43 UNION ALL
SELECT * FROM NEW_ORDERA44 UNION ALL
SELECT * FROM NEW_ORDERA45 UNION ALL
SELECT * FROM NEW_ORDERA46 UNION ALL
SELECT * FROM NEW_ORDERA47 UNION ALL
SELECT * FROM NEW_ORDERA48 UNION ALL
SELECT * FROM NEW_ORDERA49 UNION ALL
SELECT * FROM NEW_ORDERA50 UNION ALL
SELECT * FROM NEW_ORDERA51 UNION ALL
SELECT * FROM NEW_ORDERB1 UNION ALL
SELECT * FROM NEW_ORDERB2 UNION ALL
SELECT * FROM NEW_ORDERB3 UNION ALL
SELECT * FROM NEW_ORDERB4 UNION ALL
SELECT * FROM NEW_ORDERB5 UNION ALL
SELECT * FROM NEW_ORDERB6 UNION ALL
SELECT * FROM NEW_ORDERB7 UNION ALL
SELECT * FROM NEW_ORDERB8 UNION ALL
SELECT * FROM NEW_ORDERB9 UNION ALL
SELECT * FROM NEW_ORDERB10 UNION ALL
SELECT * FROM NEW_ORDERB11 UNION ALL
SELECT * FROM NEW_ORDERB12 UNION ALL
SELECT * FROM NEW_ORDERB13 UNION ALL
SELECT * FROM NEW_ORDERB14 UNION ALL
SELECT * FROM NEW_ORDERB15 UNION ALL
SELECT * FROM NEW_ORDERB16 UNION ALL
SELECT * FROM NEW_ORDERB17 UNION ALL
SELECT * FROM NEW_ORDERB18 UNION ALL
SELECT * FROM NEW_ORDERB19 UNION ALL
SELECT * FROM NEW_ORDERB20 UNION ALL
SELECT * FROM NEW_ORDERB21 UNION ALL
SELECT * FROM NEW_ORDERB22 UNION ALL
SELECT * FROM NEW_ORDERB23 UNION ALL
SELECT * FROM NEW_ORDERB24 UNION ALL

```

```

SELECT * FROM NEW_ORDERB25 UNION ALL
SELECT * FROM NEW_ORDERB26 UNION ALL
SELECT * FROM NEW_ORDERB27 UNION ALL
SELECT * FROM NEW_ORDERB28 UNION ALL
SELECT * FROM NEW_ORDERB29 UNION ALL
SELECT * FROM NEW_ORDERB30 UNION ALL
SELECT * FROM NEW_ORDERB31 UNION ALL
SELECT * FROM NEW_ORDERB32 UNION ALL
SELECT * FROM NEW_ORDERB33 UNION ALL
SELECT * FROM NEW_ORDERB34 UNION ALL
SELECT * FROM NEW_ORDERB35 UNION ALL
SELECT * FROM NEW_ORDERB36 UNION ALL
SELECT * FROM NEW_ORDERB37 UNION ALL
SELECT * FROM NEW_ORDERB38 UNION ALL
SELECT * FROM NEW_ORDERB39 UNION ALL
SELECT * FROM NEW_ORDERB40 UNION ALL
SELECT * FROM NEW_ORDERB41 UNION ALL
SELECT * FROM NEW_ORDERB42 UNION ALL
SELECT * FROM NEW_ORDERB43 UNION ALL
SELECT * FROM NEW_ORDERB44 UNION ALL
SELECT * FROM NEW_ORDERB45 UNION ALL
SELECT * FROM NEW_ORDERB46 UNION ALL
SELECT * FROM NEW_ORDERB47 UNION ALL
SELECT * FROM NEW_ORDERB48 UNION ALL
SELECT * FROM NEW_ORDERB49 UNION ALL
SELECT * FROM NEW_ORDERB50 UNION ALL
SELECT * FROM NEW_ORDERB51
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

DDL/CRVW ORDERS.ddl

```

connect to TPCC in share mode;
DROP VIEW ORDERS;
CREATE VIEW ORDERS
(O_C_ID,
O_ENTRY_D,
O_CARRIER_ID,
O_OL_CNT,
O_ALL_LOCAL,
O_ID,
O_W_ID,
O_D_ID
) AS SELECT * FROM ORDERS1 UNION ALL
SELECT * FROM ORDERS2 UNION ALL
SELECT * FROM ORDERS3 UNION ALL
SELECT * FROM ORDERS4 UNION ALL
SELECT * FROM ORDERS5 UNION ALL
SELECT * FROM ORDERS6 UNION ALL
SELECT * FROM ORDERS7 UNION ALL
SELECT * FROM ORDERS8 UNION ALL
SELECT * FROM ORDERS9 UNION ALL
SELECT * FROM ORDERS10 UNION ALL
SELECT * FROM ORDERS11 UNION ALL
SELECT * FROM ORDERS12 UNION ALL
SELECT * FROM ORDERS13 UNION ALL
SELECT * FROM ORDERS14 UNION ALL
SELECT * FROM ORDERS15 UNION ALL
SELECT * FROM ORDERS16 UNION ALL
SELECT * FROM ORDERS17 UNION ALL

```

```

SELECT * FROM ORDERS18 UNION ALL
SELECT * FROM ORDERS19 UNION ALL
SELECT * FROM ORDERS20 UNION ALL
SELECT * FROM ORDERS21 UNION ALL
SELECT * FROM ORDERS22 UNION ALL
SELECT * FROM ORDERS23 UNION ALL
SELECT * FROM ORDERS24 UNION ALL
SELECT * FROM ORDERS25 UNION ALL
SELECT * FROM ORDERS26 UNION ALL
SELECT * FROM ORDERS27 UNION ALL
SELECT * FROM ORDERS28 UNION ALL
SELECT * FROM ORDERS29 UNION ALL
SELECT * FROM ORDERS30 UNION ALL
SELECT * FROM ORDERS31 UNION ALL
SELECT * FROM ORDERS32 UNION ALL
SELECT * FROM ORDERS33 UNION ALL
SELECT * FROM ORDERS34 UNION ALL
SELECT * FROM ORDERS35 UNION ALL
SELECT * FROM ORDERS36 UNION ALL
SELECT * FROM ORDERS37 UNION ALL
SELECT * FROM ORDERS38 UNION ALL
SELECT * FROM ORDERS39 UNION ALL
SELECT * FROM ORDERS40 UNION ALL
SELECT * FROM ORDERS41 UNION ALL
SELECT * FROM ORDERS42 UNION ALL
SELECT * FROM ORDERS43 UNION ALL
SELECT * FROM ORDERS44 UNION ALL
SELECT * FROM ORDERS45 UNION ALL
SELECT * FROM ORDERS46 UNION ALL
SELECT * FROM ORDERS47 UNION ALL
SELECT * FROM ORDERS48 UNION ALL
SELECT * FROM ORDERS49 UNION ALL
SELECT * FROM ORDERS50 UNION ALL
SELECT * FROM ORDERS51
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

DDL/CRVW ORDER LINE.ddl

```

connect to TPCC in share mode;
DROP VIEW ORDER_LINE;
CREATE VIEW ORDER_LINE
(OL_DELIVERY_D,
OL_AMOUNT,
OL_I_ID,
OL_SUPPLY_W_ID,
OL_QUANTITY,
OL_DIST_INFO,
OL_O_ID,
OL_D_ID,
OL_W_ID,
OL_NUMBER
) AS SELECT * FROM ORDER_LINE1 UNION ALL
SELECT * FROM ORDER_LINE2 UNION ALL
SELECT * FROM ORDER_LINE3 UNION ALL
SELECT * FROM ORDER_LINE4 UNION ALL
SELECT * FROM ORDER_LINE5 UNION ALL
SELECT * FROM ORDER_LINE6 UNION ALL
SELECT * FROM ORDER_LINE7 UNION ALL
SELECT * FROM ORDER_LINE8 UNION ALL

```

```

SELECT * FROM ORDER_LINE9 UNION ALL
SELECT * FROM ORDER_LINE10 UNION ALL
SELECT * FROM ORDER_LINE11 UNION ALL
SELECT * FROM ORDER_LINE12 UNION ALL
SELECT * FROM ORDER_LINE13 UNION ALL
SELECT * FROM ORDER_LINE14 UNION ALL
SELECT * FROM ORDER_LINE15 UNION ALL
SELECT * FROM ORDER_LINE16 UNION ALL
SELECT * FROM ORDER_LINE17 UNION ALL
SELECT * FROM ORDER_LINE18 UNION ALL
SELECT * FROM ORDER_LINE19 UNION ALL
SELECT * FROM ORDER_LINE20 UNION ALL
SELECT * FROM ORDER_LINE21 UNION ALL
SELECT * FROM ORDER_LINE22 UNION ALL
SELECT * FROM ORDER_LINE23 UNION ALL
SELECT * FROM ORDER_LINE24 UNION ALL
SELECT * FROM ORDER_LINE25 UNION ALL
SELECT * FROM ORDER_LINE26 UNION ALL
SELECT * FROM ORDER_LINE27 UNION ALL
SELECT * FROM ORDER_LINE28 UNION ALL
SELECT * FROM ORDER_LINE29 UNION ALL
SELECT * FROM ORDER_LINE30 UNION ALL
SELECT * FROM ORDER_LINE31 UNION ALL
SELECT * FROM ORDER_LINE32 UNION ALL
SELECT * FROM ORDER_LINE33 UNION ALL
SELECT * FROM ORDER_LINE34 UNION ALL
SELECT * FROM ORDER_LINE35 UNION ALL
SELECT * FROM ORDER_LINE36 UNION ALL
SELECT * FROM ORDER_LINE37 UNION ALL
SELECT * FROM ORDER_LINE38 UNION ALL
SELECT * FROM ORDER_LINE39 UNION ALL
SELECT * FROM ORDER_LINE40 UNION ALL
SELECT * FROM ORDER_LINE41 UNION ALL
SELECT * FROM ORDER_LINE42 UNION ALL
SELECT * FROM ORDER_LINE43 UNION ALL
SELECT * FROM ORDER_LINE44 UNION ALL
SELECT * FROM ORDER_LINE45 UNION ALL
SELECT * FROM ORDER_LINE46 UNION ALL
SELECT * FROM ORDER_LINE47 UNION ALL
SELECT * FROM ORDER_LINE48 UNION ALL
SELECT * FROM ORDER_LINE49 UNION ALL
SELECT * FROM ORDER_LINE50 UNION ALL
SELECT * FROM ORDER_LINE51
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

DDL/CRVW STOCK.ddl

```

connect to TPCC in share mode;
DROP VIEW STOCK;
CREATE VIEW STOCK
(S_REMOTE_CNT,
S_QUANTITY,
S_ORDER_CNT,
S_YTD,
S_DATA,
S_DIST_01,
S_DIST_02,
S_DIST_03,
S_DIST_04,

```

```

S_DIST_05,
S_DIST_06,
S_DIST_07,
S_DIST_08,
S_DIST_09,
S_DIST_10,
S_L_ID,
S_W_ID
) AS SELECT * FROM STOCK1 UNION ALL
SELECT * FROM STOCK2 UNION ALL
SELECT * FROM STOCK3 UNION ALL
SELECT * FROM STOCK4 UNION ALL
SELECT * FROM STOCK5 UNION ALL
SELECT * FROM STOCK6 UNION ALL
SELECT * FROM STOCK7 UNION ALL
SELECT * FROM STOCK8 UNION ALL
SELECT * FROM STOCK9 UNION ALL
SELECT * FROM STOCK10 UNION ALL
SELECT * FROM STOCK11 UNION ALL
SELECT * FROM STOCK12 UNION ALL
SELECT * FROM STOCK13 UNION ALL
SELECT * FROM STOCK14 UNION ALL
SELECT * FROM STOCK15 UNION ALL
SELECT * FROM STOCK16 UNION ALL
SELECT * FROM STOCK17 UNION ALL
SELECT * FROM STOCK18 UNION ALL
SELECT * FROM STOCK19 UNION ALL
SELECT * FROM STOCK20 UNION ALL
SELECT * FROM STOCK21 UNION ALL
SELECT * FROM STOCK22 UNION ALL
SELECT * FROM STOCK23 UNION ALL
SELECT * FROM STOCK24 UNION ALL
SELECT * FROM STOCK25 UNION ALL
SELECT * FROM STOCK26 UNION ALL
SELECT * FROM STOCK27 UNION ALL
SELECT * FROM STOCK28 UNION ALL
SELECT * FROM STOCK29 UNION ALL
SELECT * FROM STOCK30 UNION ALL
SELECT * FROM STOCK31 UNION ALL
SELECT * FROM STOCK32 UNION ALL
SELECT * FROM STOCK33 UNION ALL
SELECT * FROM STOCK34 UNION ALL
SELECT * FROM STOCK35 UNION ALL
SELECT * FROM STOCK36 UNION ALL
SELECT * FROM STOCK37 UNION ALL
SELECT * FROM STOCK38 UNION ALL
SELECT * FROM STOCK39 UNION ALL
SELECT * FROM STOCK40 UNION ALL
SELECT * FROM STOCK41 UNION ALL
SELECT * FROM STOCK42 UNION ALL
SELECT * FROM STOCK43 UNION ALL
SELECT * FROM STOCK44 UNION ALL
SELECT * FROM STOCK45 UNION ALL
SELECT * FROM STOCK46 UNION ALL
SELECT * FROM STOCK47 UNION ALL
SELECT * FROM STOCK48 UNION ALL
SELECT * FROM STOCK49 UNION ALL
SELECT * FROM STOCK50 UNION ALL
SELECT * FROM STOCK51
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

DDL/CRVW WAREHOUSE.ddl

```

connect to TPCC in share mode;
DROP VIEW WAREHOUSE;
CREATE VIEW WAREHOUSE
(W_NAME,
W_STREET_1,
W_STREET_2,
W_CITY,
W_STATE,
W_ZIP,
W_TAX,
W_YTD,
W_ID
) AS SELECT * FROM WAREHOUSE1 UNION ALL
SELECT * FROM WAREHOUSE2 UNION ALL
SELECT * FROM WAREHOUSE3 UNION ALL
SELECT * FROM WAREHOUSE4 UNION ALL
SELECT * FROM WAREHOUSE5 UNION ALL
SELECT * FROM WAREHOUSE6 UNION ALL
SELECT * FROM WAREHOUSE7 UNION ALL
SELECT * FROM WAREHOUSE8 UNION ALL
SELECT * FROM WAREHOUSE9 UNION ALL
SELECT * FROM WAREHOUSE10 UNION ALL
SELECT * FROM WAREHOUSE11 UNION ALL
SELECT * FROM WAREHOUSE12 UNION ALL
SELECT * FROM WAREHOUSE13 UNION ALL
SELECT * FROM WAREHOUSE14 UNION ALL
SELECT * FROM WAREHOUSE15 UNION ALL
SELECT * FROM WAREHOUSE16 UNION ALL
SELECT * FROM WAREHOUSE17 UNION ALL
SELECT * FROM WAREHOUSE18 UNION ALL
SELECT * FROM WAREHOUSE19 UNION ALL
SELECT * FROM WAREHOUSE20 UNION ALL
SELECT * FROM WAREHOUSE21 UNION ALL
SELECT * FROM WAREHOUSE22 UNION ALL
SELECT * FROM WAREHOUSE23 UNION ALL
SELECT * FROM WAREHOUSE24 UNION ALL
SELECT * FROM WAREHOUSE25 UNION ALL
SELECT * FROM WAREHOUSE26 UNION ALL
SELECT * FROM WAREHOUSE27 UNION ALL
SELECT * FROM WAREHOUSE28 UNION ALL
SELECT * FROM WAREHOUSE29 UNION ALL
SELECT * FROM WAREHOUSE30 UNION ALL
SELECT * FROM WAREHOUSE31 UNION ALL
SELECT * FROM WAREHOUSE32 UNION ALL
SELECT * FROM WAREHOUSE33 UNION ALL
SELECT * FROM WAREHOUSE34 UNION ALL
SELECT * FROM WAREHOUSE35 UNION ALL
SELECT * FROM WAREHOUSE36 UNION ALL
SELECT * FROM WAREHOUSE37 UNION ALL
SELECT * FROM WAREHOUSE38 UNION ALL
SELECT * FROM WAREHOUSE39 UNION ALL
SELECT * FROM WAREHOUSE40 UNION ALL
SELECT * FROM WAREHOUSE41 UNION ALL
SELECT * FROM WAREHOUSE42 UNION ALL
SELECT * FROM WAREHOUSE43 UNION ALL
SELECT * FROM WAREHOUSE44 UNION ALL
SELECT * FROM WAREHOUSE45 UNION ALL
SELECT * FROM WAREHOUSE46 UNION ALL
SELECT * FROM WAREHOUSE47 UNION ALL
SELECT * FROM WAREHOUSE48 UNION ALL

```

```

SELECT * FROM WAREHOUSE49 UNION ALL
SELECT * FROM WAREHOUSE50 UNION ALL
SELECT * FROM WAREHOUSE51
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

DDL/GEN CUSTOMER ALL.sh

```

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1 800 -f1
/flats/F1_001/customer_001_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 801 1600 -f1
/flats/F1_001/customer_001_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1601 2400 -f1
/flats/F1_002/customer_002_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 2401 3200 -f1
/flats/F1_002/customer_002_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 3201 4000 -f1
/flats/F1_003/customer_003_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 4001 4800 -f1
/flats/F1_003/customer_003_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 4801 5600 -f1
/flats/F1_004/customer_004_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 5601 6400 -f1
/flats/F1_004/customer_004_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 6401 7200 -f1
/flats/F1_005/customer_005_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 7201 8000 -f1
/flats/F1_005/customer_005_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 8001 8800 -f1
/flats/F1_006/customer_006_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 8801 9600 -f1
/flats/F1_006/customer_006_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 9601 10400 -f1
/flats/F1_007/customer_007_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 10401 11200 -f1
/flats/F1_007/customer_007_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 11201 12000 -f1
/flats/F1_008/customer_008_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 12001 12800 -f1
/flats/F1_008/customer_008_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 12801 13600 -f1
/flats/F1_009/customer_009_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 13601 14400 -f1
/flats/F1_009/customer_009_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 14401 15200 -f1
/flats/F1_010/customer_010_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 15201 16000 -f1
/flats/F1_010/customer_010_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 16001 16800 -f1
/flats/F1_011/customer_011_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 16801 17600 -f1
/flats/F1_011/customer_011_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 17601 18400 -f1
/flats/F1_012/customer_012_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 18401 19200 -f1
/flats/F1_012/customer_012_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 19201 20000 -f1
/flats/F1_013/customer_013_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 20001 20800 -f1
/flats/F1_013/customer_013_2.dat

```


/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 24001 25600 -f1
/flats/F1_016/district_016_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 25601 27200 -f1
/flats/F1_017/district_017_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 27201 28800 -f1
/flats/F1_018/district_018_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 28801 30400 -f1
/flats/F1_019/district_019_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 30401 32000 -f1
/flats/F1_020/district_020_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 32001 33600 -f1
/flats/F1_021/district_021_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 33601 35200 -f1
/flats/F1_022/district_022_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 35201 36800 -f1
/flats/F1_023/district_023_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 36801 38400 -f1
/flats/F1_024/district_024_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 38401 40000 -f1
/flats/F1_025/district_025_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 40001 41600 -f1
/flats/F1_026/district_026_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 41601 43200 -f1
/flats/F1_027/district_027_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 43201 44800 -f1
/flats/F1_028/district_028_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 44801 46400 -f1
/flats/F1_029/district_029_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 46401 48000 -f1
/flats/F1_030/district_030_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 48001 49600 -f1
/flats/F1_031/district_031_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 49601 51200 -f1
/flats/F1_032/district_032_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 51201 52800 -f1
/flats/F1_033/district_033_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 52801 54400 -f1
/flats/F1_034/district_034_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 54401 56000 -f1
/flats/F1_035/district_035_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 56001 57600 -f1
/flats/F1_036/district_036_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 57601 59200 -f1
/flats/F1_037/district_037_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 59201 60800 -f1
/flats/F1_038/district_038_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 60801 62400 -f1
/flats/F1_039/district_039_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 62401 64000 -f1
/flats/F1_040/district_040_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 64001 65600 -f1
/flats/F1_041/district_041_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 65601 67200 -f1
/flats/F1_042/district_042_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 67201 68800 -f1
/flats/F1_043/district_043_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 68801 70400 -f1
/flats/F1_044/district_044_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 70401 72000 -f1
/flats/F1_045/district_045_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 72001 73600 -f1
/flats/F1_046/district_046_1.dat

/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 73601 75200 -f1
/flats/F1_047/district_047_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 75201 76800 -f1
/flats/F1_048/district_048_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 76801 78400 -f1
/flats/F1_049/district_049_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 78401 80000 -f1
/flats/F1_050/district_050_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 4 -r 80001 81600 -f1
/flats/F1_051/district_051_1.dat

DDL/GEN HISTORY ALL.sh

/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 1 1600 -f1
/flats/F1_001/history_001_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 1601 3200 -f1
/flats/F1_002/history_002_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 3201 4800 -f1
/flats/F1_003/history_003_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 4801 6400 -f1
/flats/F1_004/history_004_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 6401 8000 -f1
/flats/F1_005/history_005_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 8001 9600 -f1
/flats/F1_006/history_006_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 9601 11200 -f1
/flats/F1_007/history_007_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 11201 12800 -f1
/flats/F1_008/history_008_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 12801 14400 -f1
/flats/F1_009/history_009_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 14401 16000 -f1
/flats/F1_010/history_010_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 16001 17600 -f1
/flats/F1_011/history_011_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 17601 19200 -f1
/flats/F1_012/history_012_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 19201 20800 -f1
/flats/F1_013/history_013_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 20801 22400 -f1
/flats/F1_014/history_014_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 22401 24000 -f1
/flats/F1_015/history_015_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 24001 25600 -f1
/flats/F1_016/history_016_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 25601 27200 -f1
/flats/F1_017/history_017_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 27201 28800 -f1
/flats/F1_018/history_018_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 28801 30400 -f1
/flats/F1_019/history_019_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 30401 32000 -f1
/flats/F1_020/history_020_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 32001 33600 -f1
/flats/F1_021/history_021_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 33601 35200 -f1
/flats/F1_022/history_022_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 35201 36800 -f1
/flats/F1_023/history_023_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 36801 38400 -f1
/flats/F1_024/history_024_1.dat

/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 38401 40000 -f1
/flats/F1_025/history_025_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 40001 41600 -f1
/flats/F1_026/history_026_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 41601 43200 -f1
/flats/F1_027/history_027_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 43201 44800 -f1
/flats/F1_028/history_028_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 44801 46400 -f1
/flats/F1_029/history_029_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 46401 48000 -f1
/flats/F1_030/history_030_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 48001 49600 -f1
/flats/F1_031/history_031_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 49601 51200 -f1
/flats/F1_032/history_032_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 51201 52800 -f1
/flats/F1_033/history_033_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 52801 54400 -f1
/flats/F1_034/history_034_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 54401 56000 -f1
/flats/F1_035/history_035_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 56001 57600 -f1
/flats/F1_036/history_036_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 57601 59200 -f1
/flats/F1_037/history_037_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 59201 60800 -f1
/flats/F1_038/history_038_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 60801 62400 -f1
/flats/F1_039/history_039_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 62401 64000 -f1
/flats/F1_040/history_040_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 64001 65600 -f1
/flats/F1_041/history_041_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 65601 67200 -f1
/flats/F1_042/history_042_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 67201 68800 -f1
/flats/F1_043/history_043_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 68801 70400 -f1
/flats/F1_044/history_044_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 70401 72000 -f1
/flats/F1_045/history_045_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 72001 73600 -f1
/flats/F1_046/history_046_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 73601 75200 -f1
/flats/F1_047/history_047_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 75201 76800 -f1
/flats/F1_048/history_048_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 76801 78400 -f1
/flats/F1_049/history_049_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 78401 80000 -f1
/flats/F1_050/history_050_1.dat
/home/tpcc/tpc-c.ibm/dbgen/genedata -t 8 -r 80001 81600 -f1
/flats/F1_051/history_051_1.dat

DDL/GEN ITEM ALL.sh

/home/tpcc/tpc-c.ibm/dbgen/genedata -t 5 -f1
/flats/F1_001_001/item_001_1.dat

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 73601 74400 -f1
/flats/F1_047/neworder_047_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 74401 75200 -f1
/flats/F1_047/neworder_047_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 75201 76000 -f1
/flats/F1_048/neworder_048_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 76001 76800 -f1
/flats/F1_048/neworder_048_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 76801 77600 -f1
/flats/F1_049/neworder_049_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 77601 78400 -f1
/flats/F1_049/neworder_049_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 78401 79200 -f1
/flats/F1_050/neworder_050_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 79201 80000 -f1
/flats/F1_050/neworder_050_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 80001 80800 -f1
/flats/F1_051/neworder_051_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 80801 81600 -f1
/flats/F1_051/neworder_051_2.dat

DDL/GEN ORDERS ALL.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 1 800 -f1
/flats/F1_001/orders_001_1.dat -f2
/flats/F1_001/orderline_001_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 801 1600 -f1
/flats/F1_001/orders_001_2.dat -f2
/flats/F1_001/orderline_001_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 1601 2400 -f1
/flats/F1_002/orders_002_1.dat -f2
/flats/F1_002/orderline_002_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 2401 3200 -f1
/flats/F1_002/orders_002_2.dat -f2
/flats/F1_002/orderline_002_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 3201 4000 -f1
/flats/F1_003/orders_003_1.dat -f2
/flats/F1_003/orderline_003_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 4001 4800 -f1
/flats/F1_003/orders_003_2.dat -f2
/flats/F1_003/orderline_003_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 4801 5600 -f1
/flats/F1_004/orders_004_1.dat -f2
/flats/F1_004/orderline_004_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 5601 6400 -f1
/flats/F1_004/orders_004_2.dat -f2
/flats/F1_004/orderline_004_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 6401 7200 -f1
/flats/F1_005/orders_005_1.dat -f2
/flats/F1_005/orderline_005_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 7201 8000 -f1
/flats/F1_005/orders_005_2.dat -f2
/flats/F1_005/orderline_005_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 8001 8800 -f1
/flats/F1_006/orders_006_1.dat -f2
/flats/F1_006/orderline_006_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 8801 9600 -f1
/flats/F1_006/orders_006_2.dat -f2
/flats/F1_006/orderline_006_2.dat

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 9601 10400 -f1
/flats/F1_007/orders_007_1.dat -f2
/flats/F1_007/orderline_007_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 10401 11200 -f1
/flats/F1_007/orders_007_2.dat -f2
/flats/F1_007/orderline_007_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 11201 12000 -f1
/flats/F1_008/orders_008_1.dat -f2
/flats/F1_008/orderline_008_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 12001 12800 -f1
/flats/F1_008/orders_008_2.dat -f2
/flats/F1_008/orderline_008_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 12801 13600 -f1
/flats/F1_009/orders_009_1.dat -f2
/flats/F1_009/orderline_009_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 13601 14400 -f1
/flats/F1_009/orders_009_2.dat -f2
/flats/F1_009/orderline_009_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 14401 15200 -f1
/flats/F1_010/orders_010_1.dat -f2
/flats/F1_010/orderline_010_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 15201 16000 -f1
/flats/F1_010/orders_010_2.dat -f2
/flats/F1_010/orderline_010_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 16001 16800 -f1
/flats/F1_011/orders_011_1.dat -f2
/flats/F1_011/orderline_011_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 16801 17600 -f1
/flats/F1_011/orders_011_2.dat -f2
/flats/F1_011/orderline_011_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 17601 18400 -f1
/flats/F1_012/orders_012_1.dat -f2
/flats/F1_012/orderline_012_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 18401 19200 -f1
/flats/F1_012/orders_012_2.dat -f2
/flats/F1_012/orderline_012_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 19201 20000 -f1
/flats/F1_013/orders_013_1.dat -f2
/flats/F1_013/orderline_013_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 20001 20800 -f1
/flats/F1_013/orders_013_2.dat -f2
/flats/F1_013/orderline_013_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 20801 21600 -f1
/flats/F1_014/orders_014_1.dat -f2
/flats/F1_014/orderline_014_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 21601 22400 -f1
/flats/F1_014/orders_014_2.dat -f2
/flats/F1_014/orderline_014_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 22401 23200 -f1
/flats/F1_015/orders_015_1.dat -f2
/flats/F1_015/orderline_015_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 23201 24000 -f1
/flats/F1_015/orders_015_2.dat -f2
/flats/F1_015/orderline_015_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 24001 24800 -f1
/flats/F1_016/orders_016_1.dat -f2
/flats/F1_016/orderline_016_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 24801 25600 -f1
/flats/F1_016/orders_016_2.dat -f2
/flats/F1_016/orderline_016_2.dat

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 25601 26400 -f1
/flats/F1_017/orders_017_1.dat -f2
/flats/F1_017/orderline_017_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 26401 27200 -f1
/flats/F1_017/orders_017_2.dat -f2
/flats/F1_017/orderline_017_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 27201 28000 -f1
/flats/F1_018/orders_018_1.dat -f2
/flats/F1_018/orderline_018_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 28001 28800 -f1
/flats/F1_018/orders_018_2.dat -f2
/flats/F1_018/orderline_018_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 28801 29600 -f1
/flats/F1_019/orders_019_1.dat -f2
/flats/F1_019/orderline_019_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 29601 30400 -f1
/flats/F1_019/orders_019_2.dat -f2
/flats/F1_019/orderline_019_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 30401 31200 -f1
/flats/F1_020/orders_020_1.dat -f2
/flats/F1_020/orderline_020_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 31201 32000 -f1
/flats/F1_020/orders_020_2.dat -f2
/flats/F1_020/orderline_020_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 32001 32800 -f1
/flats/F1_021/orders_021_1.dat -f2
/flats/F1_021/orderline_021_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 32801 33600 -f1
/flats/F1_021/orders_021_2.dat -f2
/flats/F1_021/orderline_021_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 33601 34400 -f1
/flats/F1_022/orders_022_1.dat -f2
/flats/F1_022/orderline_022_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 34401 35200 -f1
/flats/F1_022/orders_022_2.dat -f2
/flats/F1_022/orderline_022_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 35201 36000 -f1
/flats/F1_023/orders_023_1.dat -f2
/flats/F1_023/orderline_023_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 36001 36800 -f1
/flats/F1_023/orders_023_2.dat -f2
/flats/F1_023/orderline_023_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 36801 37600 -f1
/flats/F1_024/orders_024_1.dat -f2
/flats/F1_024/orderline_024_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 37601 38400 -f1
/flats/F1_024/orders_024_2.dat -f2
/flats/F1_024/orderline_024_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 38401 39200 -f1
/flats/F1_025/orders_025_1.dat -f2
/flats/F1_025/orderline_025_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 39201 40000 -f1
/flats/F1_025/orders_025_2.dat -f2
/flats/F1_025/orderline_025_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 40001 40800 -f1
/flats/F1_026/orders_026_1.dat -f2
/flats/F1_026/orderline_026_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 40801 41600 -f1
/flats/F1_026/orders_026_2.dat -f2
/flats/F1_026/orderline_026_2.dat

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 41601 42400 -f1
/flats/F1_027/orders_027_1.dat -f2
/flats/F1_027/orderline_027_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 42401 43200 -f1
/flats/F1_027/orders_027_2.dat -f2
/flats/F1_027/orderline_027_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 43201 44000 -f1
/flats/F1_028/orders_028_1.dat -f2
/flats/F1_028/orderline_028_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 44001 44800 -f1
/flats/F1_028/orders_028_2.dat -f2
/flats/F1_028/orderline_028_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 44801 45600 -f1
/flats/F1_029/orders_029_1.dat -f2
/flats/F1_029/orderline_029_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 45601 46400 -f1
/flats/F1_029/orders_029_2.dat -f2
/flats/F1_029/orderline_029_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 46401 47200 -f1
/flats/F1_030/orders_030_1.dat -f2
/flats/F1_030/orderline_030_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 47201 48000 -f1
/flats/F1_030/orders_030_2.dat -f2
/flats/F1_030/orderline_030_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 48001 48800 -f1
/flats/F1_031/orders_031_1.dat -f2
/flats/F1_031/orderline_031_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 48801 49600 -f1
/flats/F1_031/orders_031_2.dat -f2
/flats/F1_031/orderline_031_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 49601 50400 -f1
/flats/F1_032/orders_032_1.dat -f2
/flats/F1_032/orderline_032_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 50401 51200 -f1
/flats/F1_032/orders_032_2.dat -f2
/flats/F1_032/orderline_032_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 51201 52000 -f1
/flats/F1_033/orders_033_1.dat -f2
/flats/F1_033/orderline_033_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 52001 52800 -f1
/flats/F1_033/orders_033_2.dat -f2
/flats/F1_033/orderline_033_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 52801 53600 -f1
/flats/F1_034/orders_034_1.dat -f2
/flats/F1_034/orderline_034_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 53601 54400 -f1
/flats/F1_034/orders_034_2.dat -f2
/flats/F1_034/orderline_034_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 54401 55200 -f1
/flats/F1_035/orders_035_1.dat -f2
/flats/F1_035/orderline_035_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 55201 56000 -f1
/flats/F1_035/orders_035_2.dat -f2
/flats/F1_035/orderline_035_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 56001 56800 -f1
/flats/F1_036/orders_036_1.dat -f2
/flats/F1_036/orderline_036_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 56801 57600 -f1
/flats/F1_036/orders_036_2.dat -f2
/flats/F1_036/orderline_036_2.dat

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 57601 58400 -f1
/flats/F1_037/orders_037_1.dat -f2
/flats/F1_037/orderline_037_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 58401 59200 -f1
/flats/F1_037/orders_037_2.dat -f2
/flats/F1_037/orderline_037_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 59201 60000 -f1
/flats/F1_038/orders_038_1.dat -f2
/flats/F1_038/orderline_038_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 60001 60800 -f1
/flats/F1_038/orders_038_2.dat -f2
/flats/F1_038/orderline_038_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 60801 61600 -f1
/flats/F1_039/orders_039_1.dat -f2
/flats/F1_039/orderline_039_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 61601 62400 -f1
/flats/F1_039/orders_039_2.dat -f2
/flats/F1_039/orderline_039_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 62401 63200 -f1
/flats/F1_040/orders_040_1.dat -f2
/flats/F1_040/orderline_040_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 63201 64000 -f1
/flats/F1_040/orders_040_2.dat -f2
/flats/F1_040/orderline_040_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 64001 64800 -f1
/flats/F1_041/orders_041_1.dat -f2
/flats/F1_041/orderline_041_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 64801 65600 -f1
/flats/F1_041/orders_041_2.dat -f2
/flats/F1_041/orderline_041_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 65601 66400 -f1
/flats/F1_042/orders_042_1.dat -f2
/flats/F1_042/orderline_042_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 66401 67200 -f1
/flats/F1_042/orders_042_2.dat -f2
/flats/F1_042/orderline_042_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 67201 68000 -f1
/flats/F1_043/orders_043_1.dat -f2
/flats/F1_043/orderline_043_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 68001 68800 -f1
/flats/F1_043/orders_043_2.dat -f2
/flats/F1_043/orderline_043_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 68801 69600 -f1
/flats/F1_044/orders_044_1.dat -f2
/flats/F1_044/orderline_044_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 69601 70400 -f1
/flats/F1_044/orders_044_2.dat -f2
/flats/F1_044/orderline_044_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 70401 71200 -f1
/flats/F1_045/orders_045_1.dat -f2
/flats/F1_045/orderline_045_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 71201 72000 -f1
/flats/F1_045/orders_045_2.dat -f2
/flats/F1_045/orderline_045_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 72001 72800 -f1
/flats/F1_046/orders_046_1.dat -f2
/flats/F1_046/orderline_046_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 72801 73600 -f1
/flats/F1_046/orders_046_2.dat -f2
/flats/F1_046/orderline_046_2.dat

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 73601 74400 -f1
/flats/F1_047/orders_047_1.dat -f2
/flats/F1_047/orderline_047_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 74401 75200 -f1
/flats/F1_047/orders_047_2.dat -f2
/flats/F1_047/orderline_047_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 75201 76000 -f1
/flats/F1_048/orders_048_1.dat -f2
/flats/F1_048/orderline_048_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 76001 76800 -f1
/flats/F1_048/orders_048_2.dat -f2
/flats/F1_048/orderline_048_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 76801 77600 -f1
/flats/F1_049/orders_049_1.dat -f2
/flats/F1_049/orderline_049_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 77601 78400 -f1
/flats/F1_049/orders_049_2.dat -f2
/flats/F1_049/orderline_049_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 78401 79200 -f1
/flats/F1_050/orders_050_1.dat -f2
/flats/F1_050/orderline_050_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 79201 80000 -f1
/flats/F1_050/orders_050_2.dat -f2
/flats/F1_050/orderline_050_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 80001 80800 -f1
/flats/F1_051/orders_051_1.dat -f2
/flats/F1_051/orderline_051_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 80801 81600 -f1
/flats/F1_051/orders_051_2.dat -f2
/flats/F1_051/orderline_051_2.dat

DDL/GEN STOCK ALL.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 1 800 -f1
/flats/F1_001/stock_001_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 801 1600 -f1
/flats/F1_001/stock_001_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 1601 2400 -f1
/flats/F1_002/stock_002_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 2401 3200 -f1
/flats/F1_002/stock_002_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 3201 4000 -f1
/flats/F1_003/stock_003_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 4001 4800 -f1
/flats/F1_003/stock_003_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 4801 5600 -f1
/flats/F1_004/stock_004_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 5601 6400 -f1
/flats/F1_004/stock_004_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 6401 7200 -f1
/flats/F1_005/stock_005_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 7201 8000 -f1
/flats/F1_005/stock_005_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 8001 8800 -f1
/flats/F1_006/stock_006_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 8801 9600 -f1
/flats/F1_006/stock_006_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 9601 10400 -f1
/flats/F1_007/stock_007_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 10401 11200 -f1
/flats/F1_007/stock_007_2.dat


```

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 4801 6400 -f1
/flats/F1_004/warehouse_004_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 6401 8000 -f1
/flats/F1_005/warehouse_005_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 8001 9600 -f1
/flats/F1_006/warehouse_006_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 9601 11200 -f1
/flats/F1_007/warehouse_007_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 11201 12800 -f1
/flats/F1_008/warehouse_008_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 12801 14400 -f1
/flats/F1_009/warehouse_009_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 14401 16000 -f1
/flats/F1_010/warehouse_010_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 16001 17600 -f1
/flats/F1_011/warehouse_011_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 17601 19200 -f1
/flats/F1_012/warehouse_012_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 19201 20800 -f1
/flats/F1_013/warehouse_013_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 20801 22400 -f1
/flats/F1_014/warehouse_014_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 22401 24000 -f1
/flats/F1_015/warehouse_015_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 24001 25600 -f1
/flats/F1_016/warehouse_016_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 25601 27200 -f1
/flats/F1_017/warehouse_017_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 27201 28800 -f1
/flats/F1_018/warehouse_018_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 28801 30400 -f1
/flats/F1_019/warehouse_019_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 30401 32000 -f1
/flats/F1_020/warehouse_020_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 32001 33600 -f1
/flats/F1_021/warehouse_021_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 33601 35200 -f1
/flats/F1_022/warehouse_022_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 35201 36800 -f1
/flats/F1_023/warehouse_023_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 36801 38400 -f1
/flats/F1_024/warehouse_024_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 38401 40000 -f1
/flats/F1_025/warehouse_025_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 40001 41600 -f1
/flats/F1_026/warehouse_026_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 41601 43200 -f1
/flats/F1_027/warehouse_027_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 43201 44800 -f1
/flats/F1_028/warehouse_028_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 44801 46400 -f1
/flats/F1_029/warehouse_029_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 46401 48000 -f1
/flats/F1_030/warehouse_030_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 48001 49600 -f1
/flats/F1_031/warehouse_031_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 49601 51200 -f1
/flats/F1_032/warehouse_032_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 51201 52800 -f1
/flats/F1_033/warehouse_033_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 52801 54400 -f1
/flats/F1_034/warehouse_034_1.dat

```

```

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 54401 56000 -f1
/flats/F1_035/warehouse_035_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 56001 57600 -f1
/flats/F1_036/warehouse_036_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 57601 59200 -f1
/flats/F1_037/warehouse_037_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 59201 60800 -f1
/flats/F1_038/warehouse_038_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 60801 62400 -f1
/flats/F1_039/warehouse_039_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 62401 64000 -f1
/flats/F1_040/warehouse_040_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 64001 65600 -f1
/flats/F1_041/warehouse_041_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 65601 67200 -f1
/flats/F1_042/warehouse_042_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 67201 68800 -f1
/flats/F1_043/warehouse_043_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 68801 70400 -f1
/flats/F1_044/warehouse_044_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 70401 72000 -f1
/flats/F1_045/warehouse_045_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 72001 73600 -f1
/flats/F1_046/warehouse_046_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 73601 75200 -f1
/flats/F1_047/warehouse_047_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 75201 76800 -f1
/flats/F1_048/warehouse_048_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 76801 78400 -f1
/flats/F1_049/warehouse_049_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 78401 80000 -f1
/flats/F1_050/warehouse_050_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 80001 81600 -f1
/flats/F1_051/warehouse_051_1.dat

```

DDL/LOAD CUSTOMER ALL.ddl

```

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER1 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_001/customer_001_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 24000000 INSERT INTO CUSTOMER1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER1 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_001/customer_001_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 24000000 INSERT INTO CUSTOMER1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER2 ACTIVATE NOT LOGGED
INITIALLY;

```

```

IMPORT FROM /flats/F1_002/customer_002_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 24000000 INSERT INTO CUSTOMER2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER2 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_002/customer_002_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 24000000 INSERT INTO CUSTOMER2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER3 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_003/customer_003_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 24000000 INSERT INTO CUSTOMER3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER3 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_003/customer_003_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 24000000 INSERT INTO CUSTOMER3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER4 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_004/customer_004_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 24000000 INSERT INTO CUSTOMER4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER4 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_004/customer_004_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 24000000 INSERT INTO CUSTOMER4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER5 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_005/customer_005_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 24000000 INSERT INTO CUSTOMER5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;

```



```

ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT39;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_040/district_040_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT40;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_041/district_041_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT41;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_042/district_042_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT42;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_043/district_043_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT43;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_044/district_044_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT44;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_045/district_045_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT45;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_046/district_046_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT46;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_047/district_047_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT47;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;

```

```

IMPORT FROM /flats/F1_048/district_048_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT48;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_049/district_049_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT49;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_050/district_050_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT50;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_051/district_051_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT51;
COMMIT WORK;
CONNECT RESET;

```

DDL/LOAD HISTORY ALL.ddl

```

connect to TPCC in share mode;
LOAD FROM /flats/F1_001/history_001_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY1 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_002/history_002_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY2 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_003/history_003_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY3 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_004/history_004_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY4 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_005/history_005_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY5 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;

```

```

LOAD FROM /flats/F1_006/history_006_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY6 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_007/history_007_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY7 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_008/history_008_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY8 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_009/history_009_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY9 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_010/history_010_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY10 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_011/history_011_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY11 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_012/history_012_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY12 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_013/history_013_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY13 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_014/history_014_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY14 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_015/history_015_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY15 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_016/history_016_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE

```



```

REPLACE INTO HISTORY47 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_048/history_048_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY48 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_049/history_049_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY49 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_050/history_050_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY50 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_051/history_051_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY51 NONRECOVERABLE DATA
BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;

```

DDL/LOAD ITEM ALL.ddl

```

CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_001/item_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 1000 INSERT INTO ITEM;
COMMIT WORK;
CONNECT RESET;

```

DDL/LOAD NEW ORDER ALL.ddl

```

CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_001/neworder_001_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_001/neworder_001_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_002/neworder_002_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;

```

```

IMPORT FROM /flats/F1_002/neworder_002_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_003/neworder_003_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_003/neworder_003_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_004/neworder_004_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_004/neworder_004_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_005/neworder_005_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_005/neworder_005_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_006/neworder_006_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_006/neworder_006_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA6;
COMMIT WORK;
CONNECT RESET;

```

```

CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_007/neworder_007_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_007/neworder_007_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_008/neworder_008_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_008/neworder_008_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_009/neworder_009_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_009/neworder_009_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_010/neworder_010_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_010/neworder_010_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_011/neworder_011_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA11;
COMMIT WORK;

```



```

COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE37 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE38 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE39 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE40 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE41 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE42 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE43 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE44 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE45 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE46 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE47 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE48 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

```

```

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE49 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE50 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE51 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

```

DDL/SORT NEW ORDER ALL.sh

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_001/neworder_001_1.dat >
/flats/F1_001/neworder_001_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_001/neworder_001_1.dat.sorted
/flats/F1_001/neworder_001_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/F1_001/neworder_001_1.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_001/neworder_001_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_001/neworder_001_2.dat >
/flats/F1_001/neworder_001_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_001/neworder_001_2.dat.sorted
/flats/F1_001/neworder_001_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/F1_001/neworder_001_2.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_001/neworder_001_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_002/neworder_002_1.dat >
/flats/F1_002/neworder_002_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_002/neworder_002_1.dat.sorted
/flats/F1_002/neworder_002_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/F1_002/neworder_002_1.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_002/neworder_002_1.dat)"
  echo "exists and that sufficient free space is available"

```

```

  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_002/neworder_002_2.dat >
/flats/F1_002/neworder_002_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_002/neworder_002_2.dat.sorted
/flats/F1_002/neworder_002_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/F1_002/neworder_002_2.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_002/neworder_002_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_003/neworder_003_1.dat >
/flats/F1_003/neworder_003_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_003/neworder_003_1.dat.sorted
/flats/F1_003/neworder_003_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/F1_003/neworder_003_1.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_003/neworder_003_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_003/neworder_003_2.dat >
/flats/F1_003/neworder_003_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_003/neworder_003_2.dat.sorted
/flats/F1_003/neworder_003_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/F1_003/neworder_003_2.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_003/neworder_003_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_004/neworder_004_1.dat >
/flats/F1_004/neworder_004_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_004/neworder_004_1.dat.sorted
/flats/F1_004/neworder_004_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/F1_004/neworder_004_1.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_004/neworder_004_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_004/neworder_004_2.dat >
/flats/F1_004/neworder_004_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_004/neworder_004_2.dat.sorted
/flats/F1_004/neworder_004_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_004/neworder_004_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_004/neworder_004_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_005/neworder_005_1.dat >
/flats/F1_005/neworder_005_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_005/neworder_005_1.dat.sorted
/flats/F1_005/neworder_005_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_005/neworder_005_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_005/neworder_005_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_005/neworder_005_2.dat >
/flats/F1_005/neworder_005_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_005/neworder_005_2.dat.sorted
/flats/F1_005/neworder_005_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_005/neworder_005_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_005/neworder_005_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_006/neworder_006_1.dat >
/flats/F1_006/neworder_006_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_006/neworder_006_1.dat.sorted
/flats/F1_006/neworder_006_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_006/neworder_006_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_006/neworder_006_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_006/neworder_006_2.dat >
/flats/F1_006/neworder_006_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_006/neworder_006_2.dat.sorted
/flats/F1_006/neworder_006_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_006/neworder_006_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_006/neworder_006_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_007/neworder_007_1.dat >
/flats/F1_007/neworder_007_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_007/neworder_007_1.dat.sorted
/flats/F1_007/neworder_007_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_007/neworder_007_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_007/neworder_007_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_007/neworder_007_2.dat >
/flats/F1_007/neworder_007_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_007/neworder_007_2.dat.sorted
/flats/F1_007/neworder_007_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_007/neworder_007_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_007/neworder_007_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_008/neworder_008_1.dat >
/flats/F1_008/neworder_008_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_008/neworder_008_1.dat.sorted
/flats/F1_008/neworder_008_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_008/neworder_008_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_008/neworder_008_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_008/neworder_008_2.dat >
/flats/F1_008/neworder_008_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_008/neworder_008_2.dat.sorted
/flats/F1_008/neworder_008_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_008/neworder_008_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_008/neworder_008_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_009/neworder_009_1.dat >
/flats/F1_009/neworder_009_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_009/neworder_009_1.dat.sorted
/flats/F1_009/neworder_009_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_009/neworder_009_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_009/neworder_009_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_009/neworder_009_2.dat >
/flats/F1_009/neworder_009_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_009/neworder_009_2.dat.sorted
/flats/F1_009/neworder_009_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_009/neworder_009_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_009/neworder_009_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_010/neworder_010_1.dat >
/flats/F1_010/neworder_010_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_010/neworder_010_1.dat.sorted
/flats/F1_010/neworder_010_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_010/neworder_010_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_010/neworder_010_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_010/neworder_010_2.dat >
/flats/F1_010/neworder_010_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_010/neworder_010_2.dat.sorted
/flats/F1_010/neworder_010_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_010/neworder_010_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_010/neworder_010_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_011/neworder_011_1.dat >
/flats/F1_011/neworder_011_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_011/neworder_011_1.dat.sorted
/flats/F1_011/neworder_011_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_011/neworder_011_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_011/neworder_011_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_011/neworder_011_2.dat >
/flats/F1_011/neworder_011_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_011/neworder_011_2.dat.sorted
/flats/F1_011/neworder_011_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_011/neworder_011_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_011/neworder_011_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_012/neworder_012_1.dat >
/flats/F1_012/neworder_012_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_012/neworder_012_1.dat.sorted
/flats/F1_012/neworder_012_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_012/neworder_012_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_012/neworder_012_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_012/neworder_012_2.dat >
/flats/F1_012/neworder_012_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_012/neworder_012_2.dat.sorted
/flats/F1_012/neworder_012_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_012/neworder_012_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_012/neworder_012_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_013/neworder_013_1.dat >
/flats/F1_013/neworder_013_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_013/neworder_013_1.dat.sorted
/flats/F1_013/neworder_013_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_013/neworder_013_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_013/neworder_013_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_013/neworder_013_2.dat >
/flats/F1_013/neworder_013_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_013/neworder_013_2.dat.sorted
/flats/F1_013/neworder_013_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_013/neworder_013_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_013/neworder_013_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_014/neworder_014_1.dat >
/flats/F1_014/neworder_014_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_014/neworder_014_1.dat.sorted
/flats/F1_014/neworder_014_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_014/neworder_014_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_014/neworder_014_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_014/neworder_014_2.dat >
/flats/F1_014/neworder_014_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_014/neworder_014_2.dat.sorted
/flats/F1_014/neworder_014_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_014/neworder_014_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_014/neworder_014_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_015/neworder_015_1.dat >
/flats/F1_015/neworder_015_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_015/neworder_015_1.dat.sorted
/flats/F1_015/neworder_015_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_015/neworder_015_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_015/neworder_015_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_015/neworder_015_2.dat >
/flats/F1_015/neworder_015_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_015/neworder_015_2.dat.sorted
/flats/F1_015/neworder_015_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_015/neworder_015_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_015/neworder_015_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_016/neworder_016_1.dat >
/flats/F1_016/neworder_016_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_016/neworder_016_1.dat.sorted
/flats/F1_016/neworder_016_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_016/neworder_016_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_016/neworder_016_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_016/neworder_016_2.dat >
/flats/F1_016/neworder_016_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_016/neworder_016_2.dat.sorted
/flats/F1_016/neworder_016_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_016/neworder_016_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_016/neworder_016_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_017/neworder_017_1.dat >
/flats/F1_017/neworder_017_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_017/neworder_017_1.dat.sorted
/flats/F1_017/neworder_017_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_017/neworder_017_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_017/neworder_017_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_017/neworder_017_2.dat >
/flats/F1_017/neworder_017_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_017/neworder_017_2.dat.sorted
/flats/F1_017/neworder_017_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_017/neworder_017_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_017/neworder_017_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_018/neworder_018_1.dat >
/flats/F1_018/neworder_018_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_018/neworder_018_1.dat.sorted
/flats/F1_018/neworder_018_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_018/neworder_018_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_018/neworder_018_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_018/neworder_018_2.dat >
/flats/F1_018/neworder_018_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_018/neworder_018_2.dat.sorted
/flats/F1_018/neworder_018_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_018/neworder_018_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_018/neworder_018_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_019/neworder_019_1.dat >
/flats/F1_019/neworder_019_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_019/neworder_019_1.dat.sorted
/flats/F1_019/neworder_019_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_019/neworder_019_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_019/neworder_019_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_019/neworder_019_2.dat >
/flats/F1_019/neworder_019_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_019/neworder_019_2.dat.sorted
/flats/F1_019/neworder_019_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_019/neworder_019_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_019/neworder_019_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_020/neworder_020_1.dat >
/flats/F1_020/neworder_020_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_020/neworder_020_1.dat.sorted
/flats/F1_020/neworder_020_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_020/neworder_020_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_020/neworder_020_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_020/neworder_020_2.dat >
/flats/F1_020/neworder_020_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_020/neworder_020_2.dat.sorted
/flats/F1_020/neworder_020_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_020/neworder_020_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_020/neworder_020_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_021/neworder_021_1.dat >
/flats/F1_021/neworder_021_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_021/neworder_021_1.dat.sorted
/flats/F1_021/neworder_021_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_021/neworder_021_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_021/neworder_021_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_021/neworder_021_2.dat >
/flats/F1_021/neworder_021_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_021/neworder_021_2.dat.sorted
/flats/F1_021/neworder_021_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_021/neworder_021_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_021/neworder_021_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_022/neworder_022_1.dat >
/flats/F1_022/neworder_022_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_022/neworder_022_1.dat.sorted
/flats/F1_022/neworder_022_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_022/neworder_022_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_022/neworder_022_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_022/neworder_022_2.dat >
/flats/F1_022/neworder_022_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_022/neworder_022_2.dat.sorted
/flats/F1_022/neworder_022_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_022/neworder_022_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_022/neworder_022_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_023/neworder_023_1.dat >
/flats/F1_023/neworder_023_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_023/neworder_023_1.dat.sorted
/flats/F1_023/neworder_023_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_023/neworder_023_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_023/neworder_023_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_023/neworder_023_2.dat >
/flats/F1_023/neworder_023_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_023/neworder_023_2.dat.sorted
/flats/F1_023/neworder_023_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_023/neworder_023_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_023/neworder_023_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_024/neworder_024_1.dat >
/flats/F1_024/neworder_024_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_024/neworder_024_1.dat.sorted
/flats/F1_024/neworder_024_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_024/neworder_024_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_024/neworder_024_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_024/neworder_024_2.dat >
/flats/F1_024/neworder_024_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_024/neworder_024_2.dat.sorted
/flats/F1_024/neworder_024_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_024/neworder_024_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_024/neworder_024_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_025/neworder_025_1.dat >
/flats/F1_025/neworder_025_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_025/neworder_025_1.dat.sorted
/flats/F1_025/neworder_025_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_025/neworder_025_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_025/neworder_025_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_025/neworder_025_2.dat >
/flats/F1_025/neworder_025_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_025/neworder_025_2.dat.sorted
/flats/F1_025/neworder_025_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_025/neworder_025_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_025/neworder_025_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_026/neworder_026_1.dat >
/flats/F1_026/neworder_026_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_026/neworder_026_1.dat.sorted
/flats/F1_026/neworder_026_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_026/neworder_026_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_026/neworder_026_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_026/neworder_026_2.dat >
/flats/F1_026/neworder_026_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_026/neworder_026_2.dat.sorted
/flats/F1_026/neworder_026_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_026/neworder_026_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_026/neworder_026_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_027/neworder_027_1.dat >
/flats/F1_027/neworder_027_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_027/neworder_027_1.dat.sorted
/flats/F1_027/neworder_027_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_027/neworder_027_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_027/neworder_027_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_027/neworder_027_2.dat >
/flats/F1_027/neworder_027_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_027/neworder_027_2.dat.sorted
/flats/F1_027/neworder_027_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_027/neworder_027_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_027/neworder_027_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_028/neworder_028_1.dat >
/flats/F1_028/neworder_028_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_028/neworder_028_1.dat.sorted
/flats/F1_028/neworder_028_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_028/neworder_028_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_028/neworder_028_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_028/neworder_028_2.dat >
/flats/F1_028/neworder_028_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_028/neworder_028_2.dat.sorted
/flats/F1_028/neworder_028_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_028/neworder_028_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_028/neworder_028_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_029/neworder_029_1.dat >
/flats/F1_029/neworder_029_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_029/neworder_029_1.dat.sorted
/flats/F1_029/neworder_029_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_029/neworder_029_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_029/neworder_029_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_029/neworder_029_2.dat >
/flats/F1_029/neworder_029_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_029/neworder_029_2.dat.sorted
/flats/F1_029/neworder_029_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_029/neworder_029_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_029/neworder_029_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_030/neworder_030_1.dat >
/flats/F1_030/neworder_030_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_030/neworder_030_1.dat.sorted
/flats/F1_030/neworder_030_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_030/neworder_030_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_030/neworder_030_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_030/neworder_030_2.dat >
/flats/F1_030/neworder_030_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_030/neworder_030_2.dat.sorted
/flats/F1_030/neworder_030_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_030/neworder_030_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_030/neworder_030_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_031/neworder_031_1.dat >
/flats/F1_031/neworder_031_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_031/neworder_031_1.dat.sorted
/flats/F1_031/neworder_031_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_031/neworder_031_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_031/neworder_031_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_031/neworder_031_2.dat >
/flats/F1_031/neworder_031_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_031/neworder_031_2.dat.sorted
/flats/F1_031/neworder_031_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_031/neworder_031_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_031/neworder_031_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_032/neworder_032_1.dat >
/flats/F1_032/neworder_032_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_032/neworder_032_1.dat.sorted
/flats/F1_032/neworder_032_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_032/neworder_032_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_032/neworder_032_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_032/neworder_032_2.dat >
/flats/F1_032/neworder_032_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_032/neworder_032_2.dat.sorted
/flats/F1_032/neworder_032_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_032/neworder_032_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_032/neworder_032_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_033/neworder_033_1.dat >
/flats/F1_033/neworder_033_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_033/neworder_033_1.dat.sorted
/flats/F1_033/neworder_033_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_033/neworder_033_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_033/neworder_033_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_033/neworder_033_2.dat >
/flats/F1_033/neworder_033_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_033/neworder_033_2.dat.sorted
/flats/F1_033/neworder_033_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_033/neworder_033_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_033/neworder_033_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_034/neworder_034_1.dat >
/flats/F1_034/neworder_034_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_034/neworder_034_1.dat.sorted
/flats/F1_034/neworder_034_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_034/neworder_034_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_034/neworder_034_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```



```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_034/neworder_034_2.dat >
/flats/F1_034/neworder_034_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_034/neworder_034_2.dat.sorted
/flats/F1_034/neworder_034_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_034/neworder_034_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_034/neworder_034_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_035/neworder_035_1.dat >
/flats/F1_035/neworder_035_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_035/neworder_035_1.dat.sorted
/flats/F1_035/neworder_035_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_035/neworder_035_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_035/neworder_035_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_035/neworder_035_2.dat >
/flats/F1_035/neworder_035_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_035/neworder_035_2.dat.sorted
/flats/F1_035/neworder_035_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_035/neworder_035_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_035/neworder_035_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_036/neworder_036_1.dat >
/flats/F1_036/neworder_036_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_036/neworder_036_1.dat.sorted
/flats/F1_036/neworder_036_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_036/neworder_036_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_036/neworder_036_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_036/neworder_036_2.dat >
/flats/F1_036/neworder_036_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_036/neworder_036_2.dat.sorted
/flats/F1_036/neworder_036_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_036/neworder_036_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_036/neworder_036_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_037/neworder_037_1.dat >
/flats/F1_037/neworder_037_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_037/neworder_037_1.dat.sorted
/flats/F1_037/neworder_037_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_037/neworder_037_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_037/neworder_037_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_037/neworder_037_2.dat >
/flats/F1_037/neworder_037_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_037/neworder_037_2.dat.sorted
/flats/F1_037/neworder_037_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_037/neworder_037_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_037/neworder_037_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_038/neworder_038_1.dat >
/flats/F1_038/neworder_038_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_038/neworder_038_1.dat.sorted
/flats/F1_038/neworder_038_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_038/neworder_038_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_038/neworder_038_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_038/neworder_038_2.dat >
/flats/F1_038/neworder_038_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_038/neworder_038_2.dat.sorted
/flats/F1_038/neworder_038_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_038/neworder_038_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_038/neworder_038_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_039/neworder_039_1.dat >
/flats/F1_039/neworder_039_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_039/neworder_039_1.dat.sorted
/flats/F1_039/neworder_039_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_039/neworder_039_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_039/neworder_039_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_039/neworder_039_2.dat >
/flats/F1_039/neworder_039_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_039/neworder_039_2.dat.sorted
/flats/F1_039/neworder_039_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_039/neworder_039_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_039/neworder_039_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_040/neworder_040_1.dat >
/flats/F1_040/neworder_040_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_040/neworder_040_1.dat.sorted
/flats/F1_040/neworder_040_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_040/neworder_040_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_040/neworder_040_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_040/neworder_040_2.dat >
/flats/F1_040/neworder_040_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_040/neworder_040_2.dat.sorted
/flats/F1_040/neworder_040_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_040/neworder_040_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_040/neworder_040_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_041/neworder_041_1.dat >
/flats/F1_041/neworder_041_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_041/neworder_041_1.dat.sorted
/flats/F1_041/neworder_041_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_041/neworder_041_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_041/neworder_041_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_041/neworder_041_2.dat >
/flats/F1_041/neworder_041_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_041/neworder_041_2.dat.sorted
/flats/F1_041/neworder_041_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_041/neworder_041_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_041/neworder_041_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_042/neworder_042_1.dat >
/flats/F1_042/neworder_042_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_042/neworder_042_1.dat.sorted
/flats/F1_042/neworder_042_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_042/neworder_042_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_042/neworder_042_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_042/neworder_042_2.dat >
/flats/F1_042/neworder_042_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_042/neworder_042_2.dat.sorted
/flats/F1_042/neworder_042_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_042/neworder_042_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_042/neworder_042_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_043/neworder_043_1.dat >
/flats/F1_043/neworder_043_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_043/neworder_043_1.dat.sorted
/flats/F1_043/neworder_043_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_043/neworder_043_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_043/neworder_043_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_043/neworder_043_2.dat >
/flats/F1_043/neworder_043_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_043/neworder_043_2.dat.sorted
/flats/F1_043/neworder_043_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_043/neworder_043_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_043/neworder_043_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_044/neworder_044_1.dat >
/flats/F1_044/neworder_044_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_044/neworder_044_1.dat.sorted
/flats/F1_044/neworder_044_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_044/neworder_044_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_044/neworder_044_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_044/neworder_044_2.dat >
/flats/F1_044/neworder_044_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_044/neworder_044_2.dat.sorted
/flats/F1_044/neworder_044_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_044/neworder_044_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_044/neworder_044_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_045/neworder_045_1.dat >
/flats/F1_045/neworder_045_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_045/neworder_045_1.dat.sorted
/flats/F1_045/neworder_045_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_045/neworder_045_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_045/neworder_045_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_045/neworder_045_2.dat >
/flats/F1_045/neworder_045_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_045/neworder_045_2.dat.sorted
/flats/F1_045/neworder_045_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_045/neworder_045_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_045/neworder_045_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_046/neworder_046_1.dat >
/flats/F1_046/neworder_046_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_046/neworder_046_1.dat.sorted
/flats/F1_046/neworder_046_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_046/neworder_046_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_046/neworder_046_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_046/neworder_046_2.dat >
/flats/F1_046/neworder_046_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_046/neworder_046_2.dat.sorted
/flats/F1_046/neworder_046_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_046/neworder_046_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_046/neworder_046_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_047/neworder_047_1.dat >
/flats/F1_047/neworder_047_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_047/neworder_047_1.dat.sorted
/flats/F1_047/neworder_047_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_047/neworder_047_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_047/neworder_047_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_047/neworder_047_2.dat >
/flats/F1_047/neworder_047_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_047/neworder_047_2.dat.sorted
/flats/F1_047/neworder_047_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_047/neworder_047_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_047/neworder_047_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_048/neworder_048_1.dat >
/flats/F1_048/neworder_048_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_048/neworder_048_1.dat.sorted
/flats/F1_048/neworder_048_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_048/neworder_048_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_048/neworder_048_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_048/neworder_048_2.dat >
/flats/F1_048/neworder_048_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_048/neworder_048_2.dat.sorted
/flats/F1_048/neworder_048_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_048/neworder_048_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_048/neworder_048_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_049/neworder_049_1.dat >
/flats/F1_049/neworder_049_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_049/neworder_049_1.dat.sorted
/flats/F1_049/neworder_049_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_049/neworder_049_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_049/neworder_049_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_049/neworder_049_2.dat >
/flats/F1_049/neworder_049_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_049/neworder_049_2.dat.sorted
/flats/F1_049/neworder_049_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_049/neworder_049_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_049/neworder_049_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_050/neworder_050_1.dat >
/flats/F1_050/neworder_050_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_050/neworder_050_1.dat.sorted
/flats/F1_050/neworder_050_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_050/neworder_050_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_050/neworder_050_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_050/neworder_050_2.dat >
/flats/F1_050/neworder_050_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_050/neworder_050_2.dat.sorted
/flats/F1_050/neworder_050_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_050/neworder_050_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_050/neworder_050_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_051/neworder_051_1.dat >
/flats/F1_051/neworder_051_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_051/neworder_051_1.dat.sorted
/flats/F1_051/neworder_051_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_051/neworder_051_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_051/neworder_051_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_051/neworder_051_2.dat >
/flats/F1_051/neworder_051_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_051/neworder_051_2.dat.sorted
/flats/F1_051/neworder_051_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_051/neworder_051_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_051/neworder_051_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

DDL/SORT ORDERS ALL.sh

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_001/orders_001_1.dat >
/flats/F1_001/orders_001_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_001/orders_001_1.dat.sorted
/flats/F1_001/orders_001_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_001/orders_001_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_001/orders_001_1.dat)"
echo "exists and that sufficient free space is available"

```

```

echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_001/orders_001_2.dat >
/flats/F1_001/orders_001_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_001/orders_001_2.dat.sorted
/flats/F1_001/orders_001_2.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_001/orders_001_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_001/orders_001_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_002/orders_002_1.dat >
/flats/F1_002/orders_002_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_002/orders_002_1.dat.sorted
/flats/F1_002/orders_002_1.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_002/orders_002_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_002/orders_002_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_002/orders_002_2.dat >
/flats/F1_002/orders_002_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_002/orders_002_2.dat.sorted
/flats/F1_002/orders_002_2.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_002/orders_002_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_002/orders_002_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_003/orders_003_1.dat >
/flats/F1_003/orders_003_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_003/orders_003_1.dat.sorted
/flats/F1_003/orders_003_1.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_003/orders_003_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_003/orders_003_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_003/orders_003_2.dat >
/flats/F1_003/orders_003_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_003/orders_003_2.dat.sorted
/flats/F1_003/orders_003_2.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_003/orders_003_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_003/orders_003_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_004/orders_004_1.dat >
/flats/F1_004/orders_004_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_004/orders_004_1.dat.sorted
/flats/F1_004/orders_004_1.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_004/orders_004_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_004/orders_004_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_004/orders_004_2.dat >
/flats/F1_004/orders_004_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_004/orders_004_2.dat.sorted
/flats/F1_004/orders_004_2.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_004/orders_004_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_004/orders_004_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_005/orders_005_1.dat >
/flats/F1_005/orders_005_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_005/orders_005_1.dat.sorted
/flats/F1_005/orders_005_1.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_005/orders_005_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_005/orders_005_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_005/orders_005_2.dat >
/flats/F1_005/orders_005_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_005/orders_005_2.dat.sorted
/flats/F1_005/orders_005_2.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_005/orders_005_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_005/orders_005_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_006/orders_006_1.dat >
/flats/F1_006/orders_006_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_006/orders_006_1.dat.sorted
/flats/F1_006/orders_006_1.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_006/orders_006_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_006/orders_006_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_006/orders_006_2.dat >
/flats/F1_006/orders_006_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_006/orders_006_2.dat.sorted
/flats/F1_006/orders_006_2.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_006/orders_006_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_006/orders_006_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_007/orders_007_1.dat >
/flats/F1_007/orders_007_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_007/orders_007_1.dat.sorted
/flats/F1_007/orders_007_1.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_007/orders_007_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_007/orders_007_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_007/orders_007_2.dat >
/flats/F1_007/orders_007_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_007/orders_007_2.dat.sorted
/flats/F1_007/orders_007_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_007/orders_007_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_007/orders_007_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_008/orders_008_1.dat >
/flats/F1_008/orders_008_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_008/orders_008_1.dat.sorted
/flats/F1_008/orders_008_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_008/orders_008_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_008/orders_008_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_008/orders_008_2.dat >
/flats/F1_008/orders_008_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_008/orders_008_2.dat.sorted
/flats/F1_008/orders_008_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_008/orders_008_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_008/orders_008_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_009/orders_009_1.dat >
/flats/F1_009/orders_009_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_009/orders_009_1.dat.sorted
/flats/F1_009/orders_009_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_009/orders_009_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_009/orders_009_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_009/orders_009_2.dat >
/flats/F1_009/orders_009_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_009/orders_009_2.dat.sorted
/flats/F1_009/orders_009_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_009/orders_009_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_009/orders_009_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_010/orders_010_1.dat >
/flats/F1_010/orders_010_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_010/orders_010_1.dat.sorted
/flats/F1_010/orders_010_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_010/orders_010_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_010/orders_010_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_010/orders_010_2.dat >
/flats/F1_010/orders_010_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_010/orders_010_2.dat.sorted
/flats/F1_010/orders_010_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_010/orders_010_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_010/orders_010_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_011/orders_011_1.dat >
/flats/F1_011/orders_011_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_011/orders_011_1.dat.sorted
/flats/F1_011/orders_011_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_011/orders_011_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_011/orders_011_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_011/orders_011_2.dat >
/flats/F1_011/orders_011_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_011/orders_011_2.dat.sorted
/flats/F1_011/orders_011_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_011/orders_011_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_011/orders_011_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_012/orders_012_1.dat >
/flats/F1_012/orders_012_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_012/orders_012_1.dat.sorted
/flats/F1_012/orders_012_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_012/orders_012_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_012/orders_012_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_012/orders_012_2.dat >
/flats/F1_012/orders_012_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_012/orders_012_2.dat.sorted
/flats/F1_012/orders_012_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_012/orders_012_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_012/orders_012_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_013/orders_013_1.dat >
/flats/F1_013/orders_013_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_013/orders_013_1.dat.sorted
/flats/F1_013/orders_013_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_013/orders_013_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_013/orders_013_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_013/orders_013_2.dat >
/flats/F1_013/orders_013_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_013/orders_013_2.dat.sorted
/flats/F1_013/orders_013_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_013/orders_013_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_013/orders_013_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_014/orders_014_1.dat >
/flats/F1_014/orders_014_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_014/orders_014_1.dat.sorted
/flats/F1_014/orders_014_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_014/orders_014_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_014/orders_014_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_014/orders_014_2.dat >
/flats/F1_014/orders_014_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_014/orders_014_2.dat.sorted
/flats/F1_014/orders_014_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_014/orders_014_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_014/orders_014_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_015/orders_015_1.dat >
/flats/F1_015/orders_015_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_015/orders_015_1.dat.sorted
/flats/F1_015/orders_015_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_015/orders_015_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_015/orders_015_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_015/orders_015_2.dat >
/flats/F1_015/orders_015_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_015/orders_015_2.dat.sorted
/flats/F1_015/orders_015_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_015/orders_015_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_015/orders_015_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_016/orders_016_1.dat >
/flats/F1_016/orders_016_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_016/orders_016_1.dat.sorted
/flats/F1_016/orders_016_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_016/orders_016_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_016/orders_016_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_016/orders_016_2.dat >
/flats/F1_016/orders_016_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_016/orders_016_2.dat.sorted
/flats/F1_016/orders_016_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_016/orders_016_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_016/orders_016_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_017/orders_017_1.dat >
/flats/F1_017/orders_017_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_017/orders_017_1.dat.sorted
/flats/F1_017/orders_017_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_017/orders_017_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_017/orders_017_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_017/orders_017_2.dat >
/flats/F1_017/orders_017_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_017/orders_017_2.dat.sorted
/flats/F1_017/orders_017_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_017/orders_017_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_017/orders_017_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_018/orders_018_1.dat >
/flats/F1_018/orders_018_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_018/orders_018_1.dat.sorted
/flats/F1_018/orders_018_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_018/orders_018_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_018/orders_018_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_018/orders_018_2.dat >
/flats/F1_018/orders_018_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_018/orders_018_2.dat.sorted
/flats/F1_018/orders_018_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_018/orders_018_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_018/orders_018_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_019/orders_019_1.dat >
/flats/F1_019/orders_019_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_019/orders_019_1.dat.sorted
/flats/F1_019/orders_019_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_019/orders_019_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_019/orders_019_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_019/orders_019_2.dat >
/flats/F1_019/orders_019_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_019/orders_019_2.dat.sorted
/flats/F1_019/orders_019_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_019/orders_019_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_019/orders_019_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_020/orders_020_1.dat >
/flats/F1_020/orders_020_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_020/orders_020_1.dat.sorted
/flats/F1_020/orders_020_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_020/orders_020_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_020/orders_020_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_020/orders_020_2.dat >
/flats/F1_020/orders_020_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_020/orders_020_2.dat.sorted
/flats/F1_020/orders_020_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_020/orders_020_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_020/orders_020_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_021/orders_021_1.dat >
/flats/F1_021/orders_021_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_021/orders_021_1.dat.sorted
/flats/F1_021/orders_021_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_021/orders_021_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_021/orders_021_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_021/orders_021_2.dat >
/flats/F1_021/orders_021_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_021/orders_021_2.dat.sorted
/flats/F1_021/orders_021_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_021/orders_021_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_021/orders_021_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_022/orders_022_1.dat >
/flats/F1_022/orders_022_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_022/orders_022_1.dat.sorted
/flats/F1_022/orders_022_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_022/orders_022_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_022/orders_022_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_022/orders_022_2.dat >
/flats/F1_022/orders_022_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_022/orders_022_2.dat.sorted
/flats/F1_022/orders_022_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_022/orders_022_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_022/orders_022_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_023/orders_023_1.dat >
/flats/F1_023/orders_023_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_023/orders_023_1.dat.sorted
/flats/F1_023/orders_023_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_023/orders_023_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_023/orders_023_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_023/orders_023_2.dat >
/flats/F1_023/orders_023_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_023/orders_023_2.dat.sorted
/flats/F1_023/orders_023_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_023/orders_023_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_023/orders_023_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_024/orders_024_1.dat >
/flats/F1_024/orders_024_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_024/orders_024_1.dat.sorted
/flats/F1_024/orders_024_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_024/orders_024_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_024/orders_024_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_024/orders_024_2.dat >
/flats/F1_024/orders_024_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_024/orders_024_2.dat.sorted
/flats/F1_024/orders_024_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_024/orders_024_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_024/orders_024_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_025/orders_025_1.dat >
/flats/F1_025/orders_025_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_025/orders_025_1.dat.sorted
/flats/F1_025/orders_025_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_025/orders_025_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_025/orders_025_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_025/orders_025_2.dat >
/flats/F1_025/orders_025_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_025/orders_025_2.dat.sorted
/flats/F1_025/orders_025_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_025/orders_025_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_025/orders_025_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_026/orders_026_1.dat >
/flats/F1_026/orders_026_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_026/orders_026_1.dat.sorted
/flats/F1_026/orders_026_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_026/orders_026_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_026/orders_026_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_026/orders_026_2.dat >
/flats/F1_026/orders_026_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_026/orders_026_2.dat.sorted
/flats/F1_026/orders_026_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_026/orders_026_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_026/orders_026_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_027/orders_027_1.dat >
/flats/F1_027/orders_027_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_027/orders_027_1.dat.sorted
/flats/F1_027/orders_027_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_027/orders_027_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_027/orders_027_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_027/orders_027_2.dat >
/flats/F1_027/orders_027_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_027/orders_027_2.dat.sorted
/flats/F1_027/orders_027_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_027/orders_027_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_027/orders_027_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_028/orders_028_1.dat >
/flats/F1_028/orders_028_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_028/orders_028_1.dat.sorted
/flats/F1_028/orders_028_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_028/orders_028_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_028/orders_028_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_028/orders_028_2.dat >
/flats/F1_028/orders_028_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_028/orders_028_2.dat.sorted
/flats/F1_028/orders_028_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_028/orders_028_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_028/orders_028_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_029/orders_029_1.dat >
/flats/F1_029/orders_029_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_029/orders_029_1.dat.sorted
/flats/F1_029/orders_029_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_029/orders_029_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_029/orders_029_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_029/orders_029_2.dat >
/flats/F1_029/orders_029_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_029/orders_029_2.dat.sorted
/flats/F1_029/orders_029_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_029/orders_029_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_029/orders_029_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_030/orders_030_1.dat >
/flats/F1_030/orders_030_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_030/orders_030_1.dat.sorted
/flats/F1_030/orders_030_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_030/orders_030_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_030/orders_030_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_030/orders_030_2.dat >
/flats/F1_030/orders_030_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_030/orders_030_2.dat.sorted
/flats/F1_030/orders_030_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_030/orders_030_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_030/orders_030_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_031/orders_031_1.dat >
/flats/F1_031/orders_031_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_031/orders_031_1.dat.sorted
/flats/F1_031/orders_031_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_031/orders_031_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_031/orders_031_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```



```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_031/orders_031_2.dat >
/flats/F1_031/orders_031_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_031/orders_031_2.dat.sorted
/flats/F1_031/orders_031_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_031/orders_031_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_031/orders_031_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_032/orders_032_1.dat >
/flats/F1_032/orders_032_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_032/orders_032_1.dat.sorted
/flats/F1_032/orders_032_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_032/orders_032_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_032/orders_032_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_032/orders_032_2.dat >
/flats/F1_032/orders_032_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_032/orders_032_2.dat.sorted
/flats/F1_032/orders_032_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_032/orders_032_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_032/orders_032_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_033/orders_033_1.dat >
/flats/F1_033/orders_033_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_033/orders_033_1.dat.sorted
/flats/F1_033/orders_033_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_033/orders_033_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_033/orders_033_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_033/orders_033_2.dat >
/flats/F1_033/orders_033_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_033/orders_033_2.dat.sorted
/flats/F1_033/orders_033_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_033/orders_033_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_033/orders_033_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_034/orders_034_1.dat >
/flats/F1_034/orders_034_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_034/orders_034_1.dat.sorted
/flats/F1_034/orders_034_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_034/orders_034_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_034/orders_034_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_034/orders_034_2.dat >
/flats/F1_034/orders_034_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_034/orders_034_2.dat.sorted
/flats/F1_034/orders_034_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_034/orders_034_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_034/orders_034_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_035/orders_035_1.dat >
/flats/F1_035/orders_035_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_035/orders_035_1.dat.sorted
/flats/F1_035/orders_035_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_035/orders_035_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_035/orders_035_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_035/orders_035_2.dat >
/flats/F1_035/orders_035_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_035/orders_035_2.dat.sorted
/flats/F1_035/orders_035_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_035/orders_035_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_035/orders_035_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_036/orders_036_1.dat >
/flats/F1_036/orders_036_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_036/orders_036_1.dat.sorted
/flats/F1_036/orders_036_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_036/orders_036_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_036/orders_036_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_036/orders_036_2.dat >
/flats/F1_036/orders_036_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_036/orders_036_2.dat.sorted
/flats/F1_036/orders_036_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_036/orders_036_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_036/orders_036_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_037/orders_037_1.dat >
/flats/F1_037/orders_037_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_037/orders_037_1.dat.sorted
/flats/F1_037/orders_037_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_037/orders_037_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_037/orders_037_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_037/orders_037_2.dat >
/flats/F1_037/orders_037_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_037/orders_037_2.dat.sorted
/flats/F1_037/orders_037_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_037/orders_037_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_037/orders_037_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_038/orders_038_1.dat >
/flats/F1_038/orders_038_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_038/orders_038_1.dat.sorted
/flats/F1_038/orders_038_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_038/orders_038_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_038/orders_038_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_038/orders_038_2.dat >
/flats/F1_038/orders_038_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_038/orders_038_2.dat.sorted
/flats/F1_038/orders_038_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_038/orders_038_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_038/orders_038_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_039/orders_039_1.dat >
/flats/F1_039/orders_039_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_039/orders_039_1.dat.sorted
/flats/F1_039/orders_039_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_039/orders_039_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_039/orders_039_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_039/orders_039_2.dat >
/flats/F1_039/orders_039_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_039/orders_039_2.dat.sorted
/flats/F1_039/orders_039_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_039/orders_039_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_039/orders_039_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_040/orders_040_1.dat >
/flats/F1_040/orders_040_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_040/orders_040_1.dat.sorted
/flats/F1_040/orders_040_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_040/orders_040_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_040/orders_040_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_040/orders_040_2.dat >
/flats/F1_040/orders_040_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_040/orders_040_2.dat.sorted
/flats/F1_040/orders_040_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_040/orders_040_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_040/orders_040_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_041/orders_041_1.dat >
/flats/F1_041/orders_041_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_041/orders_041_1.dat.sorted
/flats/F1_041/orders_041_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_041/orders_041_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_041/orders_041_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_041/orders_041_2.dat >
/flats/F1_041/orders_041_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_041/orders_041_2.dat.sorted
/flats/F1_041/orders_041_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_041/orders_041_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_041/orders_041_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_042/orders_042_1.dat >
/flats/F1_042/orders_042_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_042/orders_042_1.dat.sorted
/flats/F1_042/orders_042_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_042/orders_042_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_042/orders_042_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_042/orders_042_2.dat >
/flats/F1_042/orders_042_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_042/orders_042_2.dat.sorted
/flats/F1_042/orders_042_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_042/orders_042_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_042/orders_042_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_043/orders_043_1.dat >
/flats/F1_043/orders_043_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_043/orders_043_1.dat.sorted
/flats/F1_043/orders_043_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_043/orders_043_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_043/orders_043_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_043/orders_043_2.dat >
/flats/F1_043/orders_043_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_043/orders_043_2.dat.sorted
/flats/F1_043/orders_043_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_043/orders_043_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_043/orders_043_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_044/orders_044_1.dat >
/flats/F1_044/orders_044_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_044/orders_044_1.dat.sorted
/flats/F1_044/orders_044_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_044/orders_044_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_044/orders_044_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_044/orders_044_2.dat >
/flats/F1_044/orders_044_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_044/orders_044_2.dat.sorted
/flats/F1_044/orders_044_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_044/orders_044_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_044/orders_044_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_045/orders_045_1.dat >
/flats/F1_045/orders_045_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_045/orders_045_1.dat.sorted
/flats/F1_045/orders_045_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_045/orders_045_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_045/orders_045_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_045/orders_045_2.dat >
/flats/F1_045/orders_045_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_045/orders_045_2.dat.sorted
/flats/F1_045/orders_045_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_045/orders_045_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_045/orders_045_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_046/orders_046_1.dat >
/flats/F1_046/orders_046_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_046/orders_046_1.dat.sorted
/flats/F1_046/orders_046_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_046/orders_046_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_046/orders_046_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_046/orders_046_2.dat >
/flats/F1_046/orders_046_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_046/orders_046_2.dat.sorted
/flats/F1_046/orders_046_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_046/orders_046_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_046/orders_046_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_047/orders_047_1.dat >
/flats/F1_047/orders_047_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_047/orders_047_1.dat.sorted
/flats/F1_047/orders_047_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_047/orders_047_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_047/orders_047_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_047/orders_047_2.dat >
/flats/F1_047/orders_047_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_047/orders_047_2.dat.sorted
/flats/F1_047/orders_047_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_047/orders_047_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_047/orders_047_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_048/orders_048_1.dat >
/flats/F1_048/orders_048_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_048/orders_048_1.dat.sorted
/flats/F1_048/orders_048_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_048/orders_048_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_048/orders_048_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_048/orders_048_2.dat >
/flats/F1_048/orders_048_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_048/orders_048_2.dat.sorted
/flats/F1_048/orders_048_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_048/orders_048_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_048/orders_048_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_049/orders_049_1.dat >
/flats/F1_049/orders_049_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_049/orders_049_1.dat.sorted
/flats/F1_049/orders_049_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_049/orders_049_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_049/orders_049_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_049/orders_049_2.dat >
/flats/F1_049/orders_049_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_049/orders_049_2.dat.sorted
/flats/F1_049/orders_049_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_049/orders_049_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_049/orders_049_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_050/orders_050_1.dat >
/flats/F1_050/orders_050_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_050/orders_050_1.dat.sorted
/flats/F1_050/orders_050_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_050/orders_050_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_050/orders_050_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_050/orders_050_2.dat >
/flats/F1_050/orders_050_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_050/orders_050_2.dat.sorted
/flats/F1_050/orders_050_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_050/orders_050_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_050/orders_050_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_051/orders_051_1.dat >
/flats/F1_051/orders_051_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_051/orders_051_1.dat.sorted
/flats/F1_051/orders_051_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_051/orders_051_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_051/orders_051_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_051/orders_051_2.dat >
/flats/F1_051/orders_051_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_051/orders_051_2.dat.sorted
/flats/F1_051/orders_051_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_051/orders_051_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_051/orders_051_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

bp/alter bufferpool.ddl

```

-----
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996
-- 2006
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
-----

```

```

-- Alter Bufferpools
connect to tpcc;

alter bufferpool IBMDEFAULTBP DEFERRED size 500;
alter bufferpool IBMDEFAULTBP8K DEFERRED size 100;
alter bufferpool IBMDEFAULTBP16K DEFERRED size 100;

```

```

alter bufferpool war DEFERRED size 2587;
alter bufferpool dis DEFERRED size 23589;
alter bufferpool itm DEFERRED size 1235;
alter bufferpool hst DEFERRED size 4400;
alter bufferpool new DEFERRED size 1726367;
alter bufferpool ord DEFERRED size 1785240;
alter bufferpool ord_i DEFERRED size 5343203;
alter bufferpool oln DEFERRED size 3277877;
alter bufferpool cst DEFERRED size 7344000;
alter bufferpool cst_i DEFERRED size 2733084;
alter bufferpool stk DEFERRED size 87766856;

```

```

connect reset;
terminate;

```

bp/alter tablespace.ddl

```

-----
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International

```

```

-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996
-- 2006
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
-----

```

```

-- Set Bufferpools For Tablespaces
connect to tpcc;

```

```

-- ITEM
alter tablespace TS_ITEM_001 bufferpool ITM;

```

```

-- WAREHOUSE
alter tablespace TS_WARE_001 bufferpool WAR;
alter tablespace TS_WARE_002 bufferpool WAR;
alter tablespace TS_WARE_003 bufferpool WAR;
alter tablespace TS_WARE_004 bufferpool WAR;
alter tablespace TS_WARE_005 bufferpool WAR;
alter tablespace TS_WARE_006 bufferpool WAR;
alter tablespace TS_WARE_007 bufferpool WAR;
alter tablespace TS_WARE_008 bufferpool WAR;
alter tablespace TS_WARE_009 bufferpool WAR;
alter tablespace TS_WARE_010 bufferpool WAR;
alter tablespace TS_WARE_011 bufferpool WAR;
alter tablespace TS_WARE_012 bufferpool WAR;
alter tablespace TS_WARE_013 bufferpool WAR;
alter tablespace TS_WARE_014 bufferpool WAR;
alter tablespace TS_WARE_015 bufferpool WAR;
alter tablespace TS_WARE_016 bufferpool WAR;
alter tablespace TS_WARE_017 bufferpool WAR;
alter tablespace TS_WARE_018 bufferpool WAR;
alter tablespace TS_WARE_019 bufferpool WAR;
alter tablespace TS_WARE_020 bufferpool WAR;
alter tablespace TS_WARE_021 bufferpool WAR;
alter tablespace TS_WARE_022 bufferpool WAR;
alter tablespace TS_WARE_023 bufferpool WAR;
alter tablespace TS_WARE_024 bufferpool WAR;
alter tablespace TS_WARE_025 bufferpool WAR;
alter tablespace TS_WARE_026 bufferpool WAR;
alter tablespace TS_WARE_027 bufferpool WAR;
alter tablespace TS_WARE_028 bufferpool WAR;
alter tablespace TS_WARE_029 bufferpool WAR;
alter tablespace TS_WARE_030 bufferpool WAR;
alter tablespace TS_WARE_031 bufferpool WAR;
alter tablespace TS_WARE_032 bufferpool WAR;
alter tablespace TS_WARE_033 bufferpool WAR;
alter tablespace TS_WARE_034 bufferpool WAR;
alter tablespace TS_WARE_035 bufferpool WAR;
alter tablespace TS_WARE_036 bufferpool WAR;
alter tablespace TS_WARE_037 bufferpool WAR;
alter tablespace TS_WARE_038 bufferpool WAR;
alter tablespace TS_WARE_039 bufferpool WAR;
alter tablespace TS_WARE_040 bufferpool WAR;
alter tablespace TS_WARE_041 bufferpool WAR;
alter tablespace TS_WARE_042 bufferpool WAR;
alter tablespace TS_WARE_043 bufferpool WAR;
alter tablespace TS_WARE_044 bufferpool WAR;

```



```

-- Create Bufferpools
connect to tpcc;

create bufferpool IBMDEFAULTBP8K size 100 pagesize 8192;
create bufferpool IBMDEFAULTBP16K size 100 pagesize 16384;

create bufferpool WAR size 250 pagesize 4096;
create bufferpool DIS size 250 pagesize 4096;
create bufferpool ITM size 1000 pagesize 8192;
create bufferpool HST size 250 pagesize 16384;
create bufferpool NEW size 250 pagesize 4096;
create bufferpool ORD size 250 pagesize 8192;
create bufferpool ORD_I size 250 pagesize 8192;
create bufferpool OLN size 250 pagesize 8192;
create bufferpool CST size 250 pagesize 4096;
create bufferpool CST_I size 250 pagesize 8192;
create bufferpool STK size 100000 pagesize 4096;

connect reset;
terminate;

```

db/create_database.ddl

```

-----
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996
-- 2002
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with
-- IBM Corp.
-----
drop database tpcc;
create database tpcc on /home/tpcc/db/tpccdb1 collate using
identity
catalog tablespace
managed by system using (/home/tpcc/db/db1catalog);

```

ts/cris_customer.ddl

```

connect to tpcc;
-- now creating TS for is_customer_001 of D1

drop tablespace is_customer_001;
create regular tablespace is_customer_001 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V1CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for is_customer_002 of D1

drop tablespace is_customer_002;
create regular tablespace is_customer_002 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V2CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_003 of D1

drop tablespace is_customer_003;
create regular tablespace is_customer_003 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V3CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_004 of D1

drop tablespace is_customer_004;
create regular tablespace is_customer_004 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V4CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_005 of D1

drop tablespace is_customer_005;
create regular tablespace is_customer_005 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V5CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_006 of D1

drop tablespace is_customer_006;
create regular tablespace is_customer_006 pagesize 8K
managed by database

```

```

using
(
    device '/dev/rD1F02V1CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_007 of D1

drop tablespace is_customer_007;
create regular tablespace is_customer_007 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V2CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_008 of D1

drop tablespace is_customer_008;
create regular tablespace is_customer_008 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V3CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_009 of D1

drop tablespace is_customer_009;
create regular tablespace is_customer_009 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V4CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_010 of D1

drop tablespace is_customer_010;
create regular tablespace is_customer_010 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V5CSTI' 320192
)
extentsize 64
prefetchsize 4096

```



```

bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_011 of D1

drop tablespace is_customer_011;
create regular tablespace is_customer_011 pagesize 8K
managed by database
using
(
device '/dev/rD1F03V1CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_012 of D1

drop tablespace is_customer_012;
create regular tablespace is_customer_012 pagesize 8K
managed by database
using
(
device '/dev/rD1F03V2CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_013 of D1

drop tablespace is_customer_013;
create regular tablespace is_customer_013 pagesize 8K
managed by database
using
(
device '/dev/rD1F03V3CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_014 of D1

drop tablespace is_customer_014;
create regular tablespace is_customer_014 pagesize 8K
managed by database
using
(
device '/dev/rD1F03V4CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_015 of D1

drop tablespace is_customer_015;

```

```

create regular tablespace is_customer_015 pagesize 8K
managed by database
using
(
device '/dev/rD1F03V5CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_016 of D1

drop tablespace is_customer_016;
create regular tablespace is_customer_016 pagesize 8K
managed by database
using
(
device '/dev/rD1F04V1CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_017 of D1

drop tablespace is_customer_017;
create regular tablespace is_customer_017 pagesize 8K
managed by database
using
(
device '/dev/rD1F04V2CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_018 of D1

drop tablespace is_customer_018;
create regular tablespace is_customer_018 pagesize 8K
managed by database
using
(
device '/dev/rD1F04V3CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_019 of D1

drop tablespace is_customer_019;
create regular tablespace is_customer_019 pagesize 8K
managed by database
using
(
device '/dev/rD1F04V4CSTI' 320192
)

```

```

extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_020 of D1

drop tablespace is_customer_020;
create regular tablespace is_customer_020 pagesize 8K
managed by database
using
(
device '/dev/rD1F04V5CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_021 of D1

drop tablespace is_customer_021;
create regular tablespace is_customer_021 pagesize 8K
managed by database
using
(
device '/dev/rD1F05V1CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_022 of D1

drop tablespace is_customer_022;
create regular tablespace is_customer_022 pagesize 8K
managed by database
using
(
device '/dev/rD1F05V2CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_023 of D1

drop tablespace is_customer_023;
create regular tablespace is_customer_023 pagesize 8K
managed by database
using
(
device '/dev/rD1F05V3CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_024 of D1

```

```

drop tablespace is_customer_024;
create regular tablespace is_customer_024 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V4CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_025 of D1

drop tablespace is_customer_025;
create regular tablespace is_customer_025 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V5CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_026 of D1

drop tablespace is_customer_026;
create regular tablespace is_customer_026 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V1CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_027 of D1

drop tablespace is_customer_027;
create regular tablespace is_customer_027 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V2CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_028 of D1

drop tablespace is_customer_028;
create regular tablespace is_customer_028 pagesize 8K
managed by database
using
(

```

```

    device '/dev/rD1F06V3CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_029 of D1

drop tablespace is_customer_029;
create regular tablespace is_customer_029 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V4CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_030 of D1

drop tablespace is_customer_030;
create regular tablespace is_customer_030 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V5CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_031 of D1

drop tablespace is_customer_031;
create regular tablespace is_customer_031 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V1CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_032 of D1

drop tablespace is_customer_032;
create regular tablespace is_customer_032 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V2CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for is_customer_033 of D1

drop tablespace is_customer_033;
create regular tablespace is_customer_033 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V3CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_034 of D1

drop tablespace is_customer_034;
create regular tablespace is_customer_034 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V4CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_035 of D1

drop tablespace is_customer_035;
create regular tablespace is_customer_035 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V5CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_036 of D1

drop tablespace is_customer_036;
create regular tablespace is_customer_036 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V1CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_037 of D1

drop tablespace is_customer_037;
create regular tablespace is_customer_037 pagesize 8K
managed by database

```

```

using
(
    device '/dev/rD1F08V2CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_038 of D1

drop tablespace is_customer_038;
create regular tablespace is_customer_038 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V3CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_039 of D1

drop tablespace is_customer_039;
create regular tablespace is_customer_039 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V4CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_040 of D1

drop tablespace is_customer_040;
create regular tablespace is_customer_040 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V5CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_041 of D1

drop tablespace is_customer_041;
create regular tablespace is_customer_041 pagesize 8K
managed by database
using
(
    device '/dev/rD1F09V1CSTI' 320192
)
extentsize 64
prefetchsize 4096

```

```

bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_042 of D1

drop tablespace is_customer_042;
create regular tablespace is_customer_042 pagesize 8K
managed by database
using
(
    device '/dev/rD1F09V2CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_043 of D1

drop tablespace is_customer_043;
create regular tablespace is_customer_043 pagesize 8K
managed by database
using
(
    device '/dev/rD1F09V3CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_044 of D1

drop tablespace is_customer_044;
create regular tablespace is_customer_044 pagesize 8K
managed by database
using
(
    device '/dev/rD1F09V4CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_045 of D1

drop tablespace is_customer_045;
create regular tablespace is_customer_045 pagesize 8K
managed by database
using
(
    device '/dev/rD1F09V5CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_046 of D1

drop tablespace is_customer_046;

```

```

create regular tablespace is_customer_046 pagesize 8K
managed by database
using
(
    device '/dev/rD1F10V1CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_047 of D1

drop tablespace is_customer_047;
create regular tablespace is_customer_047 pagesize 8K
managed by database
using
(
    device '/dev/rD1F10V2CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_048 of D1

drop tablespace is_customer_048;
create regular tablespace is_customer_048 pagesize 8K
managed by database
using
(
    device '/dev/rD1F10V3CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_049 of D1

drop tablespace is_customer_049;
create regular tablespace is_customer_049 pagesize 8K
managed by database
using
(
    device '/dev/rD1F10V4CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_050 of D1

drop tablespace is_customer_050;
create regular tablespace is_customer_050 pagesize 8K
managed by database
using
(
    device '/dev/rD1F10V5CSTI' 320192
)

```

```

    extentsize 64
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_051 of D1

drop tablespace is_customer_051;
create regular tablespace is_customer_051 pagesize 8K
managed by database
using
(
    device '/dev/rD1F11V1CSTI' 320192
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

connect reset;

```

ts/cris_order.ddl

```

connect to tpc;
-- now creating TS for is_order_001 of D1

drop tablespace is_order_001;
create regular tablespace is_order_001 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V1ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_002 of D1

drop tablespace is_order_002;
create regular tablespace is_order_002 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V2ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_003 of D1

drop tablespace is_order_003;
create regular tablespace is_order_003 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V3ORDI' 258432

```

```

)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_004 of D1

drop tablespace is_order_004;
create regular tablespace is_order_004 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V4ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_005 of D1

drop tablespace is_order_005;
create regular tablespace is_order_005 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V5ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_006 of D1

drop tablespace is_order_006;
create regular tablespace is_order_006 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V1ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_007 of D1

drop tablespace is_order_007;
create regular tablespace is_order_007 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V2ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for is_order_008 of D1

drop tablespace is_order_008;
create regular tablespace is_order_008 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V3ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_009 of D1

drop tablespace is_order_009;
create regular tablespace is_order_009 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V4ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_010 of D1

drop tablespace is_order_010;
create regular tablespace is_order_010 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V5ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_011 of D1

drop tablespace is_order_011;
create regular tablespace is_order_011 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V1ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_012 of D1

drop tablespace is_order_012;
create regular tablespace is_order_012 pagesize 8K
managed by database
using

```

```

(
    device '/dev/rD1F03V2ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_013 of D1

drop tablespace is_order_013;
create regular tablespace is_order_013 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V3ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_014 of D1

drop tablespace is_order_014;
create regular tablespace is_order_014 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V4ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_015 of D1

drop tablespace is_order_015;
create regular tablespace is_order_015 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V5ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_016 of D1

drop tablespace is_order_016;
create regular tablespace is_order_016 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V1ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;

```

```

commit;

-- now creating TS for is_order_017 of D1

drop tablespace is_order_017;
create regular tablespace is_order_017 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V2ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_018 of D1

drop tablespace is_order_018;
create regular tablespace is_order_018 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V3ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_019 of D1

drop tablespace is_order_019;
create regular tablespace is_order_019 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V4ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_020 of D1

drop tablespace is_order_020;
create regular tablespace is_order_020 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V5ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_021 of D1

drop tablespace is_order_021;
create regular tablespace is_order_021 pagesize 8K

```

```

managed by database
using
(
    device '/dev/rD1F05V1ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_022 of D1

drop tablespace is_order_022;
create regular tablespace is_order_022 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V2ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_023 of D1

drop tablespace is_order_023;
create regular tablespace is_order_023 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V3ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_024 of D1

drop tablespace is_order_024;
create regular tablespace is_order_024 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V4ORDI' 258432
)
extentsize 64
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_025 of D1

drop tablespace is_order_025;
create regular tablespace is_order_025 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V5ORDI' 258432
)
extentsize 64

```

```

    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_026 of D1

drop tablespace is_order_026;
create regular tablespace is_order_026 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V1ORDI' 258432
)
    extentsize 64
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_027 of D1

drop tablespace is_order_027;
create regular tablespace is_order_027 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V2ORDI' 258432
)
    extentsize 64
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_028 of D1

drop tablespace is_order_028;
create regular tablespace is_order_028 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V3ORDI' 258432
)
    extentsize 64
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_029 of D1

drop tablespace is_order_029;
create regular tablespace is_order_029 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V4ORDI' 258432
)
    extentsize 64
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_030 of D1

```

```

drop tablespace is_order_030;
create regular tablespace is_order_030 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V5ORDI' 258432
)
    extentsize 64
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_031 of D1

drop tablespace is_order_031;
create regular tablespace is_order_031 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V1ORDI' 258432
)
    extentsize 64
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_032 of D1

drop tablespace is_order_032;
create regular tablespace is_order_032 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V2ORDI' 258432
)
    extentsize 64
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_033 of D1

drop tablespace is_order_033;
create regular tablespace is_order_033 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V3ORDI' 258432
)
    extentsize 64
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_034 of D1

drop tablespace is_order_034;
create regular tablespace is_order_034 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V4ORDI' 258432
)

```

```

)
    extentsize 64
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_035 of D1

drop tablespace is_order_035;
create regular tablespace is_order_035 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V5ORDI' 258432
)
    extentsize 64
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_036 of D1

drop tablespace is_order_036;
create regular tablespace is_order_036 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V1ORDI' 258432
)
    extentsize 64
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_037 of D1

drop tablespace is_order_037;
create regular tablespace is_order_037 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V2ORDI' 258432
)
    extentsize 64
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_038 of D1

drop tablespace is_order_038;
create regular tablespace is_order_038 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V3ORDI' 258432
)
    extentsize 64
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for is_order_039 of D1

drop tablespace is_order_039;
create regular tablespace is_order_039 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F08V4ORDI' 258432
  )
  extentsize 64
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_040 of D1

drop tablespace is_order_040;
create regular tablespace is_order_040 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F08V5ORDI' 258432
  )
  extentsize 64
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_041 of D1

drop tablespace is_order_041;
create regular tablespace is_order_041 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F09V1ORDI' 258432
  )
  extentsize 64
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_042 of D1

drop tablespace is_order_042;
create regular tablespace is_order_042 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F09V2ORDI' 258432
  )
  extentsize 64
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_043 of D1

drop tablespace is_order_043;
create regular tablespace is_order_043 pagesize 8K
  managed by database
  using

```

```

  (
    device '/dev/rD1F09V3ORDI' 258432
  )
  extentsize 64
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_044 of D1

drop tablespace is_order_044;
create regular tablespace is_order_044 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F09V4ORDI' 258432
  )
  extentsize 64
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_045 of D1

drop tablespace is_order_045;
create regular tablespace is_order_045 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F09V5ORDI' 258432
  )
  extentsize 64
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_046 of D1

drop tablespace is_order_046;
create regular tablespace is_order_046 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F10V1ORDI' 258432
  )
  extentsize 64
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_047 of D1

drop tablespace is_order_047;
create regular tablespace is_order_047 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F10V2ORDI' 258432
  )
  extentsize 64
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;

```

```

commit;

-- now creating TS for is_order_048 of D1

drop tablespace is_order_048;
create regular tablespace is_order_048 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F10V3ORDI' 258432
  )
  extentsize 64
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_049 of D1

drop tablespace is_order_049;
create regular tablespace is_order_049 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F10V4ORDI' 258432
  )
  extentsize 64
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_050 of D1

drop tablespace is_order_050;
create regular tablespace is_order_050 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F10V5ORDI' 258432
  )
  extentsize 64
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_051 of D1

drop tablespace is_order_051;
create regular tablespace is_order_051 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F11V1ORDI' 258432
  )
  extentsize 64
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

connect reset;

```

ts/crts_customer.ddl

```
connect to tpcc;
-- now creating TS for ts_customer_001 of D1

drop tablespace ts_customer_001;
create regular tablespace ts_customer_001 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_002 of D1

drop tablespace ts_customer_002;
create regular tablespace ts_customer_002 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_003 of D1

drop tablespace ts_customer_003;
create regular tablespace ts_customer_003 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_004 of D1

drop tablespace ts_customer_004;
create regular tablespace ts_customer_004 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_005 of D1

drop tablespace ts_customer_005;
create regular tablespace ts_customer_005 pagesize 4K
managed by database
using
(
```

```
    device '/dev/rD1F01V5CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_006 of D1

drop tablespace ts_customer_006;
create regular tablespace ts_customer_006 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_007 of D1

drop tablespace ts_customer_007;
create regular tablespace ts_customer_007 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_008 of D1

drop tablespace ts_customer_008;
create regular tablespace ts_customer_008 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_009 of D1

drop tablespace ts_customer_009;
create regular tablespace ts_customer_009 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_010 of D1

drop tablespace ts_customer_010;
create regular tablespace ts_customer_010 pagesize 4K
```

```
managed by database
using
(
    device '/dev/rD1F02V5CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_011 of D1

drop tablespace ts_customer_011;
create regular tablespace ts_customer_011 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_012 of D1

drop tablespace ts_customer_012;
create regular tablespace ts_customer_012 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_013 of D1

drop tablespace ts_customer_013;
create regular tablespace ts_customer_013 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_014 of D1

drop tablespace ts_customer_014;
create regular tablespace ts_customer_014 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_015 of D1
```



```

drop tablespace ts_customer_015;
create regular tablespace ts_customer_015 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V5CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_016 of D1

drop tablespace ts_customer_016;
create regular tablespace ts_customer_016 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_017 of D1

drop tablespace ts_customer_017;
create regular tablespace ts_customer_017 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_018 of D1

drop tablespace ts_customer_018;
create regular tablespace ts_customer_018 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_019 of D1

drop tablespace ts_customer_019;
create regular tablespace ts_customer_019 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V4CST' 10081536
)
extentsize 256
prefetchsize 4096;

```

```

commit;

-- now creating TS for ts_customer_020 of D1

drop tablespace ts_customer_020;
create regular tablespace ts_customer_020 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V5CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_021 of D1

drop tablespace ts_customer_021;
create regular tablespace ts_customer_021 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_022 of D1

drop tablespace ts_customer_022;
create regular tablespace ts_customer_022 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_023 of D1

drop tablespace ts_customer_023;
create regular tablespace ts_customer_023 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_024 of D1

drop tablespace ts_customer_024;
create regular tablespace ts_customer_024 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V4CST' 10081536

```

```

)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_025 of D1

drop tablespace ts_customer_025;
create regular tablespace ts_customer_025 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V5CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_026 of D1

drop tablespace ts_customer_026;
create regular tablespace ts_customer_026 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_027 of D1

drop tablespace ts_customer_027;
create regular tablespace ts_customer_027 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_028 of D1

drop tablespace ts_customer_028;
create regular tablespace ts_customer_028 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_029 of D1

drop tablespace ts_customer_029;
create regular tablespace ts_customer_029 pagesize 4K
managed by database

```

```

using
(
    device '/dev/rD1F06V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_030 of D1

drop tablespace ts_customer_030;
create regular tablespace ts_customer_030 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V5CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_031 of D1

drop tablespace ts_customer_031;
create regular tablespace ts_customer_031 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_032 of D1

drop tablespace ts_customer_032;
create regular tablespace ts_customer_032 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_033 of D1

drop tablespace ts_customer_033;
create regular tablespace ts_customer_033 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_034 of D1

```

```

drop tablespace ts_customer_034;
create regular tablespace ts_customer_034 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_035 of D1

drop tablespace ts_customer_035;
create regular tablespace ts_customer_035 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V5CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_036 of D1

drop tablespace ts_customer_036;
create regular tablespace ts_customer_036 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_037 of D1

drop tablespace ts_customer_037;
create regular tablespace ts_customer_037 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_038 of D1

drop tablespace ts_customer_038;
create regular tablespace ts_customer_038 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_customer_039 of D1

drop tablespace ts_customer_039;
create regular tablespace ts_customer_039 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_040 of D1

drop tablespace ts_customer_040;
create regular tablespace ts_customer_040 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V5CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_041 of D1

drop tablespace ts_customer_041;
create regular tablespace ts_customer_041 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_042 of D1

drop tablespace ts_customer_042;
create regular tablespace ts_customer_042 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_043 of D1

drop tablespace ts_customer_043;
create regular tablespace ts_customer_043 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V3CST' 10081536
)

```

```

        extentsize 256
        prefetchsize 4096;
commit;

-- now creating TS for ts_customer_044 of D1

drop tablespace ts_customer_044;
create regular tablespace ts_customer_044 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_045 of D1

drop tablespace ts_customer_045;
create regular tablespace ts_customer_045 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V5CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_046 of D1

drop tablespace ts_customer_046;
create regular tablespace ts_customer_046 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_047 of D1

drop tablespace ts_customer_047;
create regular tablespace ts_customer_047 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_048 of D1

drop tablespace ts_customer_048;
create regular tablespace ts_customer_048 pagesize 4K
managed by database
using

```

```

(
    device '/dev/rD1F10V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_049 of D1

drop tablespace ts_customer_049;
create regular tablespace ts_customer_049 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_050 of D1

drop tablespace ts_customer_050;
create regular tablespace ts_customer_050 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V5CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_051 of D1

drop tablespace ts_customer_051;
create regular tablespace ts_customer_051 pagesize 4K
managed by database
using
(
    device '/dev/rD1F11V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;

connect reset;

ts/crts_dist.ddl

connect to tpcc;
-- now creating TS for ts_dist_001 of D1

drop tablespace ts_dist_001;
create regular tablespace ts_dist_001 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V1DIST' 2048
)

```

```

        extentsize 256
        prefetchsize 4096;
commit;

-- now creating TS for ts_dist_002 of D1

drop tablespace ts_dist_002;
create regular tablespace ts_dist_002 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V2DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_003 of D1

drop tablespace ts_dist_003;
create regular tablespace ts_dist_003 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V3DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_004 of D1

drop tablespace ts_dist_004;
create regular tablespace ts_dist_004 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V4DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_005 of D1

drop tablespace ts_dist_005;
create regular tablespace ts_dist_005 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V5DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_006 of D1

drop tablespace ts_dist_006;
create regular tablespace ts_dist_006 pagesize 4K
managed by database
using

```

```

(
    device '/dev/rD1F02V1DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_007 of D1

drop tablespace ts_dist_007;
create regular tablespace ts_dist_007 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V2DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_008 of D1

drop tablespace ts_dist_008;
create regular tablespace ts_dist_008 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V3DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_009 of D1

drop tablespace ts_dist_009;
create regular tablespace ts_dist_009 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V4DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_010 of D1

drop tablespace ts_dist_010;
create regular tablespace ts_dist_010 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V5DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_011 of D1

drop tablespace ts_dist_011;

```

```

create regular tablespace ts_dist_011 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V1DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_012 of D1

drop tablespace ts_dist_012;
create regular tablespace ts_dist_012 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V2DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_013 of D1

drop tablespace ts_dist_013;
create regular tablespace ts_dist_013 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V3DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_014 of D1

drop tablespace ts_dist_014;
create regular tablespace ts_dist_014 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V4DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_015 of D1

drop tablespace ts_dist_015;
create regular tablespace ts_dist_015 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V5DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_dist_016 of D1

drop tablespace ts_dist_016;
create regular tablespace ts_dist_016 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V1DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_017 of D1

drop tablespace ts_dist_017;
create regular tablespace ts_dist_017 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V2DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_018 of D1

drop tablespace ts_dist_018;
create regular tablespace ts_dist_018 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V3DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_019 of D1

drop tablespace ts_dist_019;
create regular tablespace ts_dist_019 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V4DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_020 of D1

drop tablespace ts_dist_020;
create regular tablespace ts_dist_020 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V5DIST' 2048
)
extentsize 256

```

```

        prefetchsize 4096;
commit;

-- now creating TS for ts_dist_021 of D1

drop tablespace ts_dist_021;
create regular tablespace ts_dist_021 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V1DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_022 of D1

drop tablespace ts_dist_022;
create regular tablespace ts_dist_022 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V2DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_023 of D1

drop tablespace ts_dist_023;
create regular tablespace ts_dist_023 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V3DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_024 of D1

drop tablespace ts_dist_024;
create regular tablespace ts_dist_024 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V4DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_025 of D1

drop tablespace ts_dist_025;
create regular tablespace ts_dist_025 pagesize 4K
managed by database
using
(

```

```

        device '/dev/rD1F05V5DIST' 2048
    )
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_dist_026 of D1

drop tablespace ts_dist_026;
create regular tablespace ts_dist_026 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V1DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_027 of D1

drop tablespace ts_dist_027;
create regular tablespace ts_dist_027 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V2DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_028 of D1

drop tablespace ts_dist_028;
create regular tablespace ts_dist_028 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V3DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_029 of D1

drop tablespace ts_dist_029;
create regular tablespace ts_dist_029 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V4DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_030 of D1

drop tablespace ts_dist_030;
create regular tablespace ts_dist_030 pagesize 4K

```

```

managed by database
using
(
    device '/dev/rD1F06V5DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_031 of D1

drop tablespace ts_dist_031;
create regular tablespace ts_dist_031 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V1DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_032 of D1

drop tablespace ts_dist_032;
create regular tablespace ts_dist_032 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V2DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_033 of D1

drop tablespace ts_dist_033;
create regular tablespace ts_dist_033 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V3DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_034 of D1

drop tablespace ts_dist_034;
create regular tablespace ts_dist_034 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V4DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_035 of D1

```

```

drop tablespace ts_dist_035;
create regular tablespace ts_dist_035 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V5DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_036 of D1

drop tablespace ts_dist_036;
create regular tablespace ts_dist_036 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V1DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_037 of D1

drop tablespace ts_dist_037;
create regular tablespace ts_dist_037 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V2DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_038 of D1

drop tablespace ts_dist_038;
create regular tablespace ts_dist_038 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V3DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_039 of D1

drop tablespace ts_dist_039;
create regular tablespace ts_dist_039 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V4DIST' 2048
)
extentsize 256
prefetchsize 4096;

```

```

commit;

-- now creating TS for ts_dist_040 of D1

drop tablespace ts_dist_040;
create regular tablespace ts_dist_040 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V5DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_041 of D1

drop tablespace ts_dist_041;
create regular tablespace ts_dist_041 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V1DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_042 of D1

drop tablespace ts_dist_042;
create regular tablespace ts_dist_042 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V2DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_043 of D1

drop tablespace ts_dist_043;
create regular tablespace ts_dist_043 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V3DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_044 of D1

drop tablespace ts_dist_044;
create regular tablespace ts_dist_044 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V4DIST' 2048

```

```

)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_045 of D1

drop tablespace ts_dist_045;
create regular tablespace ts_dist_045 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V5DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_046 of D1

drop tablespace ts_dist_046;
create regular tablespace ts_dist_046 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V1DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_047 of D1

drop tablespace ts_dist_047;
create regular tablespace ts_dist_047 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V2DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_048 of D1

drop tablespace ts_dist_048;
create regular tablespace ts_dist_048 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V3DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_049 of D1

drop tablespace ts_dist_049;
create regular tablespace ts_dist_049 pagesize 4K
managed by database

```

```

using
(
  device '/dev/rD1F10V4DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_050 of D1

drop tablespace ts_dist_050;
create regular tablespace ts_dist_050 pagesize 4K
managed by database
using
(
  device '/dev/rD1F10V5DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_051 of D1

drop tablespace ts_dist_051;
create regular tablespace ts_dist_051 pagesize 4K
managed by database
using
(
  device '/dev/rD1F11V1DIST' 2048
)
extentsize 256
prefetchsize 4096;
commit;

connect reset;

```

ts/crts_history.ddl

```

connect to tpcc;
-- now creating TS for ts_history_001 of D1

drop tablespace ts_history_001;
create regular tablespace ts_history_001 pagesize 16K
managed by database
using
(
  device '/dev/rD1F01V1HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_002 of D1

drop tablespace ts_history_002;
create regular tablespace ts_history_002 pagesize 16K
managed by database
using
(

```

```

  device '/dev/rD1F01V2HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_003 of D1

drop tablespace ts_history_003;
create regular tablespace ts_history_003 pagesize 16K
managed by database
using
(
  device '/dev/rD1F01V3HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_004 of D1

drop tablespace ts_history_004;
create regular tablespace ts_history_004 pagesize 16K
managed by database
using
(
  device '/dev/rD1F01V4HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_005 of D1

drop tablespace ts_history_005;
create regular tablespace ts_history_005 pagesize 16K
managed by database
using
(
  device '/dev/rD1F01V5HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_006 of D1

drop tablespace ts_history_006;
create regular tablespace ts_history_006 pagesize 16K
managed by database
using
(
  device '/dev/rD1F02V1HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

```

```

-- now creating TS for ts_history_007 of D1

drop tablespace ts_history_007;
create regular tablespace ts_history_007 pagesize 16K
managed by database
using
(
  device '/dev/rD1F02V2HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_008 of D1

drop tablespace ts_history_008;
create regular tablespace ts_history_008 pagesize 16K
managed by database
using
(
  device '/dev/rD1F02V3HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_009 of D1

drop tablespace ts_history_009;
create regular tablespace ts_history_009 pagesize 16K
managed by database
using
(
  device '/dev/rD1F02V4HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_010 of D1

drop tablespace ts_history_010;
create regular tablespace ts_history_010 pagesize 16K
managed by database
using
(
  device '/dev/rD1F02V5HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_011 of D1

drop tablespace ts_history_011;
create regular tablespace ts_history_011 pagesize 16K
managed by database

```

```

using
(
    device '/dev/rD1F03V1HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_012 of D1

drop tablespace ts_history_012;
create regular tablespace ts_history_012 pagesize 16K
managed by database
using
(
    device '/dev/rD1F03V2HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_013 of D1

drop tablespace ts_history_013;
create regular tablespace ts_history_013 pagesize 16K
managed by database
using
(
    device '/dev/rD1F03V3HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_014 of D1

drop tablespace ts_history_014;
create regular tablespace ts_history_014 pagesize 16K
managed by database
using
(
    device '/dev/rD1F03V4HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_015 of D1

drop tablespace ts_history_015;
create regular tablespace ts_history_015 pagesize 16K
managed by database
using
(
    device '/dev/rD1F03V5HIST' 290000
)
extentsize 256
prefetchsize 4096

```

```

bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_016 of D1

drop tablespace ts_history_016;
create regular tablespace ts_history_016 pagesize 16K
managed by database
using
(
    device '/dev/rD1F04V1HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_017 of D1

drop tablespace ts_history_017;
create regular tablespace ts_history_017 pagesize 16K
managed by database
using
(
    device '/dev/rD1F04V2HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_018 of D1

drop tablespace ts_history_018;
create regular tablespace ts_history_018 pagesize 16K
managed by database
using
(
    device '/dev/rD1F04V3HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_019 of D1

drop tablespace ts_history_019;
create regular tablespace ts_history_019 pagesize 16K
managed by database
using
(
    device '/dev/rD1F04V4HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_020 of D1

drop tablespace ts_history_020;

```

```

create regular tablespace ts_history_020 pagesize 16K
managed by database
using
(
    device '/dev/rD1F04V5HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_021 of D1

drop tablespace ts_history_021;
create regular tablespace ts_history_021 pagesize 16K
managed by database
using
(
    device '/dev/rD1F05V1HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_022 of D1

drop tablespace ts_history_022;
create regular tablespace ts_history_022 pagesize 16K
managed by database
using
(
    device '/dev/rD1F05V2HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_023 of D1

drop tablespace ts_history_023;
create regular tablespace ts_history_023 pagesize 16K
managed by database
using
(
    device '/dev/rD1F05V3HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_024 of D1

drop tablespace ts_history_024;
create regular tablespace ts_history_024 pagesize 16K
managed by database
using
(
    device '/dev/rD1F05V4HIST' 290000
)

```



```

    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_025 of D1

drop tablespace ts_history_025;
create regular tablespace ts_history_025 pagesize 16K
managed by database
using
(
    device '/dev/rD1F05V5HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_026 of D1

drop tablespace ts_history_026;
create regular tablespace ts_history_026 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V1HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_027 of D1

drop tablespace ts_history_027;
create regular tablespace ts_history_027 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V2HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_028 of D1

drop tablespace ts_history_028;
create regular tablespace ts_history_028 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V3HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_029 of D1

```

```

drop tablespace ts_history_029;
create regular tablespace ts_history_029 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V4HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_030 of D1

drop tablespace ts_history_030;
create regular tablespace ts_history_030 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V5HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_031 of D1

drop tablespace ts_history_031;
create regular tablespace ts_history_031 pagesize 16K
managed by database
using
(
    device '/dev/rD1F07V1HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_032 of D1

drop tablespace ts_history_032;
create regular tablespace ts_history_032 pagesize 16K
managed by database
using
(
    device '/dev/rD1F07V2HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_033 of D1

drop tablespace ts_history_033;
create regular tablespace ts_history_033 pagesize 16K
managed by database
using
(

```

```

    device '/dev/rD1F07V3HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_034 of D1

drop tablespace ts_history_034;
create regular tablespace ts_history_034 pagesize 16K
managed by database
using
(
    device '/dev/rD1F07V4HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_035 of D1

drop tablespace ts_history_035;
create regular tablespace ts_history_035 pagesize 16K
managed by database
using
(
    device '/dev/rD1F07V5HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_036 of D1

drop tablespace ts_history_036;
create regular tablespace ts_history_036 pagesize 16K
managed by database
using
(
    device '/dev/rD1F08V1HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_037 of D1

drop tablespace ts_history_037;
create regular tablespace ts_history_037 pagesize 16K
managed by database
using
(
    device '/dev/rD1F08V2HIST' 290000
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

```

```

-- now creating TS for ts_history_038 of D1

drop tablespace ts_history_038;
create regular tablespace ts_history_038 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F08V3HIST' 290000
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_039 of D1

drop tablespace ts_history_039;
create regular tablespace ts_history_039 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F08V4HIST' 290000
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_040 of D1

drop tablespace ts_history_040;
create regular tablespace ts_history_040 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F08V5HIST' 290000
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_041 of D1

drop tablespace ts_history_041;
create regular tablespace ts_history_041 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F09V1HIST' 290000
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_042 of D1

drop tablespace ts_history_042;
create regular tablespace ts_history_042 pagesize 16K
  managed by database

```

```

  using
  (
    device '/dev/rD1F09V2HIST' 290000
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_043 of D1

drop tablespace ts_history_043;
create regular tablespace ts_history_043 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F09V3HIST' 290000
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_044 of D1

drop tablespace ts_history_044;
create regular tablespace ts_history_044 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F09V4HIST' 290000
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_045 of D1

drop tablespace ts_history_045;
create regular tablespace ts_history_045 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F09V5HIST' 290000
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_046 of D1

drop tablespace ts_history_046;
create regular tablespace ts_history_046 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F10V1HIST' 290000
  )
  extentsize 256
  prefetchsize 4096

```

```

  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_047 of D1

drop tablespace ts_history_047;
create regular tablespace ts_history_047 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F10V2HIST' 290000
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_048 of D1

drop tablespace ts_history_048;
create regular tablespace ts_history_048 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F10V3HIST' 290000
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_049 of D1

drop tablespace ts_history_049;
create regular tablespace ts_history_049 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F10V4HIST' 290000
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_050 of D1

drop tablespace ts_history_050;
create regular tablespace ts_history_050 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F10V5HIST' 290000
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_051 of D1

drop tablespace ts_history_051;

```

```

create regular tablespace ts_history_051 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F11V1HIST' 290000
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp16K;
commit;

connect reset;

```

ts/crts_item.ddl

```

connect to tpcc;
-- now creating TS for ts_item_001 of D1

drop tablespace ts_item_001;
create regular tablespace ts_item_001 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F01V1ITEM' 304,
    device '/dev/rD1F01V2ITEM' 304,
    device '/dev/rD1F01V3ITEM' 304,
    device '/dev/rD1F01V4ITEM' 304,
    device '/dev/rD1F01V5ITEM' 304,
    device '/dev/rD1F02V1ITEM' 304,
    device '/dev/rD1F02V2ITEM' 304,
    device '/dev/rD1F02V3ITEM' 304,
    device '/dev/rD1F02V4ITEM' 304,
    device '/dev/rD1F02V5ITEM' 304,
    device '/dev/rD1F03V1ITEM' 304,
    device '/dev/rD1F03V2ITEM' 304,
    device '/dev/rD1F03V3ITEM' 304,
    device '/dev/rD1F03V4ITEM' 304,
    device '/dev/rD1F03V5ITEM' 304,
    device '/dev/rD1F04V1ITEM' 304,
    device '/dev/rD1F04V2ITEM' 304,
    device '/dev/rD1F04V3ITEM' 304,
    device '/dev/rD1F04V4ITEM' 304,
    device '/dev/rD1F04V5ITEM' 304,
    device '/dev/rD1F05V1ITEM' 304,
    device '/dev/rD1F05V2ITEM' 304,
    device '/dev/rD1F05V3ITEM' 304,
    device '/dev/rD1F05V4ITEM' 304,
    device '/dev/rD1F05V5ITEM' 304,
    device '/dev/rD1F06V1ITEM' 304,
    device '/dev/rD1F06V2ITEM' 304,
    device '/dev/rD1F06V3ITEM' 304,
    device '/dev/rD1F06V4ITEM' 304,
    device '/dev/rD1F06V5ITEM' 304,
    device '/dev/rD1F07V1ITEM' 304,
    device '/dev/rD1F07V2ITEM' 304,
    device '/dev/rD1F07V3ITEM' 304,
    device '/dev/rD1F07V4ITEM' 304,
    device '/dev/rD1F07V5ITEM' 304,
    device '/dev/rD1F08V1ITEM' 304,
    device '/dev/rD1F08V2ITEM' 304,

```

```

    device '/dev/rD1F08V3ITEM' 304,
    device '/dev/rD1F08V4ITEM' 304,
    device '/dev/rD1F08V5ITEM' 304,
    device '/dev/rD1F09V1ITEM' 304,
    device '/dev/rD1F09V2ITEM' 304,
    device '/dev/rD1F09V3ITEM' 304,
    device '/dev/rD1F09V4ITEM' 304,
    device '/dev/rD1F09V5ITEM' 304,
    device '/dev/rD1F10V1ITEM' 304,
    device '/dev/rD1F10V2ITEM' 304,
    device '/dev/rD1F10V3ITEM' 304,
    device '/dev/rD1F10V4ITEM' 304,
    device '/dev/rD1F10V5ITEM' 304
  )
  extentsize 16
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

connect reset;

```

ts/crts_neworda.ddl

```

connect to tpcc;
-- now creating TS for ts_neworda_001 of D1

drop tablespace ts_neworda_001;
create regular tablespace ts_neworda_001 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V1NORA' 150016
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_002 of D1

drop tablespace ts_neworda_002;
create regular tablespace ts_neworda_002 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V2NORA' 150016
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_003 of D1

drop tablespace ts_neworda_003;
create regular tablespace ts_neworda_003 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V3NORA' 150016
  )
  extentsize 256

```

```

  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_004 of D1

drop tablespace ts_neworda_004;
create regular tablespace ts_neworda_004 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V4NORA' 150016
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_005 of D1

drop tablespace ts_neworda_005;
create regular tablespace ts_neworda_005 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V5NORA' 150016
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_006 of D1

drop tablespace ts_neworda_006;
create regular tablespace ts_neworda_006 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F02V1NORA' 150016
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_007 of D1

drop tablespace ts_neworda_007;
create regular tablespace ts_neworda_007 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F02V2NORA' 150016
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_008 of D1

drop tablespace ts_neworda_008;
create regular tablespace ts_neworda_008 pagesize 4K
  managed by database
  using
  (

```

```

        device '/dev/rD1F02V3NORA' 150016
    )
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_009 of D1

drop tablespace ts_neworda_009;
create regular tablespace ts_neworda_009 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V4NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_010 of D1

drop tablespace ts_neworda_010;
create regular tablespace ts_neworda_010 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V5NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_011 of D1

drop tablespace ts_neworda_011;
create regular tablespace ts_neworda_011 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V1NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_012 of D1

drop tablespace ts_neworda_012;
create regular tablespace ts_neworda_012 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V2NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_013 of D1

drop tablespace ts_neworda_013;
create regular tablespace ts_neworda_013 pagesize 4K

```

```

managed by database
using
(
    device '/dev/rD1F03V3NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_014 of D1

drop tablespace ts_neworda_014;
create regular tablespace ts_neworda_014 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V4NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_015 of D1

drop tablespace ts_neworda_015;
create regular tablespace ts_neworda_015 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V5NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_016 of D1

drop tablespace ts_neworda_016;
create regular tablespace ts_neworda_016 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V1NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_017 of D1

drop tablespace ts_neworda_017;
create regular tablespace ts_neworda_017 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V2NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_018 of D1

```

```

drop tablespace ts_neworda_018;
create regular tablespace ts_neworda_018 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V3NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_019 of D1

drop tablespace ts_neworda_019;
create regular tablespace ts_neworda_019 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V4NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_020 of D1

drop tablespace ts_neworda_020;
create regular tablespace ts_neworda_020 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V5NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_021 of D1

drop tablespace ts_neworda_021;
create regular tablespace ts_neworda_021 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V1NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_022 of D1

drop tablespace ts_neworda_022;
create regular tablespace ts_neworda_022 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V2NORA' 150016
)
extentsize 256
prefetchsize 4096;

```

```

commit;

-- now creating TS for ts_neworda_023 of D1

drop tablespace ts_neworda_023;
create regular tablespace ts_neworda_023 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V3NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_024 of D1

drop tablespace ts_neworda_024;
create regular tablespace ts_neworda_024 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V4NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_025 of D1

drop tablespace ts_neworda_025;
create regular tablespace ts_neworda_025 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V5NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_026 of D1

drop tablespace ts_neworda_026;
create regular tablespace ts_neworda_026 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V1NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_027 of D1

drop tablespace ts_neworda_027;
create regular tablespace ts_neworda_027 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V2NORA' 150016

```

```

)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_028 of D1

drop tablespace ts_neworda_028;
create regular tablespace ts_neworda_028 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V3NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_029 of D1

drop tablespace ts_neworda_029;
create regular tablespace ts_neworda_029 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V4NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_030 of D1

drop tablespace ts_neworda_030;
create regular tablespace ts_neworda_030 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V5NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_031 of D1

drop tablespace ts_neworda_031;
create regular tablespace ts_neworda_031 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V1NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_032 of D1

drop tablespace ts_neworda_032;
create regular tablespace ts_neworda_032 pagesize 4K
managed by database

```

```

using
(
    device '/dev/rD1F07V2NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_033 of D1

drop tablespace ts_neworda_033;
create regular tablespace ts_neworda_033 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V3NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_034 of D1

drop tablespace ts_neworda_034;
create regular tablespace ts_neworda_034 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V4NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_035 of D1

drop tablespace ts_neworda_035;
create regular tablespace ts_neworda_035 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V5NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_036 of D1

drop tablespace ts_neworda_036;
create regular tablespace ts_neworda_036 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V1NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_037 of D1

```

```

drop tablespace ts_neworda_037;
create regular tablespace ts_neworda_037 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V2NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_038 of D1

drop tablespace ts_neworda_038;
create regular tablespace ts_neworda_038 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V3NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_039 of D1

drop tablespace ts_neworda_039;
create regular tablespace ts_neworda_039 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V4NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_040 of D1

drop tablespace ts_neworda_040;
create regular tablespace ts_neworda_040 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V5NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_041 of D1

drop tablespace ts_neworda_041;
create regular tablespace ts_neworda_041 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V1NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_neworda_042 of D1

drop tablespace ts_neworda_042;
create regular tablespace ts_neworda_042 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V2NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_043 of D1

drop tablespace ts_neworda_043;
create regular tablespace ts_neworda_043 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V3NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_044 of D1

drop tablespace ts_neworda_044;
create regular tablespace ts_neworda_044 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V4NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_045 of D1

drop tablespace ts_neworda_045;
create regular tablespace ts_neworda_045 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V5NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_046 of D1

drop tablespace ts_neworda_046;
create regular tablespace ts_neworda_046 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V1NORA' 150016
)

```

```

    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_047 of D1

drop tablespace ts_neworda_047;
create regular tablespace ts_neworda_047 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V2NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_048 of D1

drop tablespace ts_neworda_048;
create regular tablespace ts_neworda_048 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V3NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_049 of D1

drop tablespace ts_neworda_049;
create regular tablespace ts_neworda_049 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V4NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_050 of D1

drop tablespace ts_neworda_050;
create regular tablespace ts_neworda_050 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V5NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_051 of D1

drop tablespace ts_neworda_051;
create regular tablespace ts_neworda_051 pagesize 4K
managed by database
using

```

```

(
    device '/dev/rD1F11V1NORA' 150016
)
extentsize 256
prefetchsize 4096;
commit;

connect reset;

ts/crts_newordb.ddl

connect to tpcc;
-- now creating TS for ts_newordb_001 of D1

drop tablespace ts_newordb_001;
create regular tablespace ts_newordb_001 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V1NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_002 of D1

drop tablespace ts_newordb_002;
create regular tablespace ts_newordb_002 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V2NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_003 of D1

drop tablespace ts_newordb_003;
create regular tablespace ts_newordb_003 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V3NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_004 of D1

drop tablespace ts_newordb_004;
create regular tablespace ts_newordb_004 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V4NORB' 150016
)

```

```

extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_005 of D1

drop tablespace ts_newordb_005;
create regular tablespace ts_newordb_005 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V5NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_006 of D1

drop tablespace ts_newordb_006;
create regular tablespace ts_newordb_006 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V1NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_007 of D1

drop tablespace ts_newordb_007;
create regular tablespace ts_newordb_007 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V2NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_008 of D1

drop tablespace ts_newordb_008;
create regular tablespace ts_newordb_008 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V3NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_009 of D1

drop tablespace ts_newordb_009;
create regular tablespace ts_newordb_009 pagesize 4K
managed by database
using

```

```

(
    device '/dev/rD1F02V4NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_010 of D1

drop tablespace ts_newordb_010;
create regular tablespace ts_newordb_010 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V5NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_011 of D1

drop tablespace ts_newordb_011;
create regular tablespace ts_newordb_011 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V1NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_012 of D1

drop tablespace ts_newordb_012;
create regular tablespace ts_newordb_012 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V2NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_013 of D1

drop tablespace ts_newordb_013;
create regular tablespace ts_newordb_013 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V3NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_014 of D1

drop tablespace ts_newordb_014;

```

```

create regular tablespace ts_newordb_014 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V4NORB' 150016
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_015 of D1

drop tablespace ts_newordb_015;
create regular tablespace ts_newordb_015 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V5NORB' 150016
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_016 of D1

drop tablespace ts_newordb_016;
create regular tablespace ts_newordb_016 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V1NORB' 150016
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_017 of D1

drop tablespace ts_newordb_017;
create regular tablespace ts_newordb_017 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V2NORB' 150016
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_018 of D1

drop tablespace ts_newordb_018;
create regular tablespace ts_newordb_018 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V3NORB' 150016
  )
  extentsize 256
  prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_newordb_019 of D1

drop tablespace ts_newordb_019;
create regular tablespace ts_newordb_019 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V4NORB' 150016
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_020 of D1

drop tablespace ts_newordb_020;
create regular tablespace ts_newordb_020 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V5NORB' 150016
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_021 of D1

drop tablespace ts_newordb_021;
create regular tablespace ts_newordb_021 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V1NORB' 150016
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_022 of D1

drop tablespace ts_newordb_022;
create regular tablespace ts_newordb_022 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V2NORB' 150016
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_023 of D1

drop tablespace ts_newordb_023;
create regular tablespace ts_newordb_023 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V3NORB' 150016
  )
  extentsize 256

```

```

  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_024 of D1

drop tablespace ts_newordb_024;
create regular tablespace ts_newordb_024 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V4NORB' 150016
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_025 of D1

drop tablespace ts_newordb_025;
create regular tablespace ts_newordb_025 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V5NORB' 150016
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_026 of D1

drop tablespace ts_newordb_026;
create regular tablespace ts_newordb_026 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V1NORB' 150016
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_027 of D1

drop tablespace ts_newordb_027;
create regular tablespace ts_newordb_027 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V2NORB' 150016
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_028 of D1

drop tablespace ts_newordb_028;
create regular tablespace ts_newordb_028 pagesize 4K
  managed by database
  using
  (

```



```

        device '/dev/rD1F06V3NORB' 150016
    )
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_029 of D1

drop tablespace ts_newordb_029;
create regular tablespace ts_newordb_029 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V4NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_030 of D1

drop tablespace ts_newordb_030;
create regular tablespace ts_newordb_030 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V5NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_031 of D1

drop tablespace ts_newordb_031;
create regular tablespace ts_newordb_031 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V1NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_032 of D1

drop tablespace ts_newordb_032;
create regular tablespace ts_newordb_032 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V2NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_033 of D1

drop tablespace ts_newordb_033;
create regular tablespace ts_newordb_033 pagesize 4K

```

```

managed by database
using
(
    device '/dev/rD1F07V3NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_034 of D1

drop tablespace ts_newordb_034;
create regular tablespace ts_newordb_034 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V4NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_035 of D1

drop tablespace ts_newordb_035;
create regular tablespace ts_newordb_035 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V5NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_036 of D1

drop tablespace ts_newordb_036;
create regular tablespace ts_newordb_036 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V1NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_037 of D1

drop tablespace ts_newordb_037;
create regular tablespace ts_newordb_037 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V2NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_038 of D1

```

```

drop tablespace ts_newordb_038;
create regular tablespace ts_newordb_038 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V3NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_039 of D1

drop tablespace ts_newordb_039;
create regular tablespace ts_newordb_039 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V4NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_040 of D1

drop tablespace ts_newordb_040;
create regular tablespace ts_newordb_040 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V5NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_041 of D1

drop tablespace ts_newordb_041;
create regular tablespace ts_newordb_041 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V1NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_042 of D1

drop tablespace ts_newordb_042;
create regular tablespace ts_newordb_042 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V2NORB' 150016
)
extentsize 256
prefetchsize 4096;

```

```

commit;

-- now creating TS for ts_newordb_043 of D1

drop tablespace ts_newordb_043;
create regular tablespace ts_newordb_043 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V3NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_044 of D1

drop tablespace ts_newordb_044;
create regular tablespace ts_newordb_044 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V4NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_045 of D1

drop tablespace ts_newordb_045;
create regular tablespace ts_newordb_045 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V5NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_046 of D1

drop tablespace ts_newordb_046;
create regular tablespace ts_newordb_046 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V1NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_047 of D1

drop tablespace ts_newordb_047;
create regular tablespace ts_newordb_047 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V2NORB' 150016

```

```

)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_048 of D1

drop tablespace ts_newordb_048;
create regular tablespace ts_newordb_048 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V3NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_049 of D1

drop tablespace ts_newordb_049;
create regular tablespace ts_newordb_049 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V4NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_050 of D1

drop tablespace ts_newordb_050;
create regular tablespace ts_newordb_050 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V5NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_051 of D1

drop tablespace ts_newordb_051;
create regular tablespace ts_newordb_051 pagesize 4K
managed by database
using
(
    device '/dev/rD1F11V1NORB' 150016
)
extentsize 256
prefetchsize 4096;
commit;

connect reset;

```

ts/crts_order.ddl

```

connect to tpcc;
-- now creating TS for ts_order_001 of D1

drop tablespace ts_order_001;
create regular tablespace ts_order_001 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V1ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_002 of D1

drop tablespace ts_order_002;
create regular tablespace ts_order_002 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V2ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_003 of D1

drop tablespace ts_order_003;
create regular tablespace ts_order_003 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V3ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_004 of D1

drop tablespace ts_order_004;
create regular tablespace ts_order_004 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V4ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_005 of D1

drop tablespace ts_order_005;

```

```

create regular tablespace ts_order_005 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F01V5ORD' 291584
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_006 of D1

drop tablespace ts_order_006;
create regular tablespace ts_order_006 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F02V1ORD' 291584
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_007 of D1

drop tablespace ts_order_007;
create regular tablespace ts_order_007 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F02V2ORD' 291584
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_008 of D1

drop tablespace ts_order_008;
create regular tablespace ts_order_008 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F02V3ORD' 291584
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_009 of D1

drop tablespace ts_order_009;
create regular tablespace ts_order_009 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F02V4ORD' 291584
  )

```

```

  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_010 of D1

drop tablespace ts_order_010;
create regular tablespace ts_order_010 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F02V5ORD' 291584
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_011 of D1

drop tablespace ts_order_011;
create regular tablespace ts_order_011 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V1ORD' 291584
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_012 of D1

drop tablespace ts_order_012;
create regular tablespace ts_order_012 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V2ORD' 291584
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_013 of D1

drop tablespace ts_order_013;
create regular tablespace ts_order_013 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V3ORD' 291584
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_014 of D1

```

```

drop tablespace ts_order_014;
create regular tablespace ts_order_014 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V4ORD' 291584
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_015 of D1

drop tablespace ts_order_015;
create regular tablespace ts_order_015 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V5ORD' 291584
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_016 of D1

drop tablespace ts_order_016;
create regular tablespace ts_order_016 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V1ORD' 291584
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_017 of D1

drop tablespace ts_order_017;
create regular tablespace ts_order_017 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V2ORD' 291584
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_018 of D1

drop tablespace ts_order_018;
create regular tablespace ts_order_018 pagesize 8K
  managed by database
  using
  (

```

```

        device '/dev/rD1F04V3ORD' 291584
    )
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_019 of D1

drop tablespace ts_order_019;
create regular tablespace ts_order_019 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V4ORD' 291584
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_020 of D1

drop tablespace ts_order_020;
create regular tablespace ts_order_020 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V5ORD' 291584
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_021 of D1

drop tablespace ts_order_021;
create regular tablespace ts_order_021 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V1ORD' 291584
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_022 of D1

drop tablespace ts_order_022;
create regular tablespace ts_order_022 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V2ORD' 291584
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for ts_order_023 of D1

drop tablespace ts_order_023;
create regular tablespace ts_order_023 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V3ORD' 291584
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_024 of D1

drop tablespace ts_order_024;
create regular tablespace ts_order_024 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V4ORD' 291584
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_025 of D1

drop tablespace ts_order_025;
create regular tablespace ts_order_025 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V5ORD' 291584
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_026 of D1

drop tablespace ts_order_026;
create regular tablespace ts_order_026 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V1ORD' 291584
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_027 of D1

drop tablespace ts_order_027;
create regular tablespace ts_order_027 pagesize 8K
managed by database

```

```

using
(
    device '/dev/rD1F06V2ORD' 291584
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_028 of D1

drop tablespace ts_order_028;
create regular tablespace ts_order_028 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V3ORD' 291584
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_029 of D1

drop tablespace ts_order_029;
create regular tablespace ts_order_029 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V4ORD' 291584
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_030 of D1

drop tablespace ts_order_030;
create regular tablespace ts_order_030 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V5ORD' 291584
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_031 of D1

drop tablespace ts_order_031;
create regular tablespace ts_order_031 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V1ORD' 291584
)
    extentsize 256
    prefetchsize 4096

```

```

bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_032 of D1

drop tablespace ts_order_032;
create regular tablespace ts_order_032 pagesize 8K
managed by database
using
(
device '/dev/rD1F07V2ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_033 of D1

drop tablespace ts_order_033;
create regular tablespace ts_order_033 pagesize 8K
managed by database
using
(
device '/dev/rD1F07V3ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_034 of D1

drop tablespace ts_order_034;
create regular tablespace ts_order_034 pagesize 8K
managed by database
using
(
device '/dev/rD1F07V4ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_035 of D1

drop tablespace ts_order_035;
create regular tablespace ts_order_035 pagesize 8K
managed by database
using
(
device '/dev/rD1F07V5ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_036 of D1

drop tablespace ts_order_036;

```

```

create regular tablespace ts_order_036 pagesize 8K
managed by database
using
(
device '/dev/rD1F08V1ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_037 of D1

drop tablespace ts_order_037;
create regular tablespace ts_order_037 pagesize 8K
managed by database
using
(
device '/dev/rD1F08V2ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_038 of D1

drop tablespace ts_order_038;
create regular tablespace ts_order_038 pagesize 8K
managed by database
using
(
device '/dev/rD1F08V3ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_039 of D1

drop tablespace ts_order_039;
create regular tablespace ts_order_039 pagesize 8K
managed by database
using
(
device '/dev/rD1F08V4ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_040 of D1

drop tablespace ts_order_040;
create regular tablespace ts_order_040 pagesize 8K
managed by database
using
(
device '/dev/rD1F08V5ORD' 291584
)

```

```

extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_041 of D1

drop tablespace ts_order_041;
create regular tablespace ts_order_041 pagesize 8K
managed by database
using
(
device '/dev/rD1F09V1ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_042 of D1

drop tablespace ts_order_042;
create regular tablespace ts_order_042 pagesize 8K
managed by database
using
(
device '/dev/rD1F09V2ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_043 of D1

drop tablespace ts_order_043;
create regular tablespace ts_order_043 pagesize 8K
managed by database
using
(
device '/dev/rD1F09V3ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_044 of D1

drop tablespace ts_order_044;
create regular tablespace ts_order_044 pagesize 8K
managed by database
using
(
device '/dev/rD1F09V4ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_045 of D1

```

```

drop tablespace ts_order_045;
create regular tablespace ts_order_045 pagesize 8K
managed by database
using
(
    device '/dev/rD1F09V5ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_046 of D1

drop tablespace ts_order_046;
create regular tablespace ts_order_046 pagesize 8K
managed by database
using
(
    device '/dev/rD1F10V1ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_047 of D1

drop tablespace ts_order_047;
create regular tablespace ts_order_047 pagesize 8K
managed by database
using
(
    device '/dev/rD1F10V2ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_048 of D1

drop tablespace ts_order_048;
create regular tablespace ts_order_048 pagesize 8K
managed by database
using
(
    device '/dev/rD1F10V3ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_049 of D1

drop tablespace ts_order_049;
create regular tablespace ts_order_049 pagesize 8K
managed by database
using
(

```

```

        device '/dev/rD1F10V4ORD' 291584
    )
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_050 of D1

drop tablespace ts_order_050;
create regular tablespace ts_order_050 pagesize 8K
managed by database
using
(
    device '/dev/rD1F10V5ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_051 of D1

drop tablespace ts_order_051;
create regular tablespace ts_order_051 pagesize 8K
managed by database
using
(
    device '/dev/rD1F11V1ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

connect reset;

```

ts/crts_orderline.ddl

```

connect to tpcc;
-- now creating TS for ts_orderline_001 of D1

drop tablespace ts_orderline_001;
create regular tablespace ts_orderline_001 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V1ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_002 of D1

drop tablespace ts_orderline_002;
create regular tablespace ts_orderline_002 pagesize 8K
managed by database
using

```

```

(
    device '/dev/rD1F01V2ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_003 of D1

drop tablespace ts_orderline_003;
create regular tablespace ts_orderline_003 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V3ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_004 of D1

drop tablespace ts_orderline_004;
create regular tablespace ts_orderline_004 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V4ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_005 of D1

drop tablespace ts_orderline_005;
create regular tablespace ts_orderline_005 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V5ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_006 of D1

drop tablespace ts_orderline_006;
create regular tablespace ts_orderline_006 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V1ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;

```

```

commit;

-- now creating TS for ts_orderline_007 of D1

drop tablespace ts_orderline_007;
create regular tablespace ts_orderline_007 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V2ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_008 of D1

drop tablespace ts_orderline_008;
create regular tablespace ts_orderline_008 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V3ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_009 of D1

drop tablespace ts_orderline_009;
create regular tablespace ts_orderline_009 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V4ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_010 of D1

drop tablespace ts_orderline_010;
create regular tablespace ts_orderline_010 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V5ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_011 of D1

drop tablespace ts_orderline_011;
create regular tablespace ts_orderline_011 pagesize 8K

```

```

managed by database
using
(
    device '/dev/rD1F03V1ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_012 of D1

drop tablespace ts_orderline_012;
create regular tablespace ts_orderline_012 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V2ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_013 of D1

drop tablespace ts_orderline_013;
create regular tablespace ts_orderline_013 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V3ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_014 of D1

drop tablespace ts_orderline_014;
create regular tablespace ts_orderline_014 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V4ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_015 of D1

drop tablespace ts_orderline_015;
create regular tablespace ts_orderline_015 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V5ORL' 7783936
)
extentsize 256

```

```

prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_016 of D1

drop tablespace ts_orderline_016;
create regular tablespace ts_orderline_016 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V1ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_017 of D1

drop tablespace ts_orderline_017;
create regular tablespace ts_orderline_017 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V2ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_018 of D1

drop tablespace ts_orderline_018;
create regular tablespace ts_orderline_018 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V3ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_019 of D1

drop tablespace ts_orderline_019;
create regular tablespace ts_orderline_019 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V4ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_020 of D1

```

```

drop tablespace ts_orderline_020;
create regular tablespace ts_orderline_020 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V5ORL' 7783936
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_021 of D1

drop tablespace ts_orderline_021;
create regular tablespace ts_orderline_021 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V1ORL' 7783936
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_022 of D1

drop tablespace ts_orderline_022;
create regular tablespace ts_orderline_022 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V2ORL' 7783936
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_023 of D1

drop tablespace ts_orderline_023;
create regular tablespace ts_orderline_023 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V3ORL' 7783936
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_024 of D1

drop tablespace ts_orderline_024;
create regular tablespace ts_orderline_024 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V4ORL' 7783936

```

```

    )
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_025 of D1

drop tablespace ts_orderline_025;
create regular tablespace ts_orderline_025 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V5ORL' 7783936
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_026 of D1

drop tablespace ts_orderline_026;
create regular tablespace ts_orderline_026 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V1ORL' 7783936
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_027 of D1

drop tablespace ts_orderline_027;
create regular tablespace ts_orderline_027 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V2ORL' 7783936
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_028 of D1

drop tablespace ts_orderline_028;
create regular tablespace ts_orderline_028 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V3ORL' 7783936
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for ts_orderline_029 of D1

drop tablespace ts_orderline_029;
create regular tablespace ts_orderline_029 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V4ORL' 7783936
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_030 of D1

drop tablespace ts_orderline_030;
create regular tablespace ts_orderline_030 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V5ORL' 7783936
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_031 of D1

drop tablespace ts_orderline_031;
create regular tablespace ts_orderline_031 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V1ORL' 7783936
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_032 of D1

drop tablespace ts_orderline_032;
create regular tablespace ts_orderline_032 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V2ORL' 7783936
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_033 of D1

drop tablespace ts_orderline_033;
create regular tablespace ts_orderline_033 pagesize 8K
  managed by database
  using

```



```

(
    device '/dev/rD1F07V3ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_034 of D1

drop tablespace ts_orderline_034;
create regular tablespace ts_orderline_034 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V4ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_035 of D1

drop tablespace ts_orderline_035;
create regular tablespace ts_orderline_035 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V5ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_036 of D1

drop tablespace ts_orderline_036;
create regular tablespace ts_orderline_036 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V1ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_037 of D1

drop tablespace ts_orderline_037;
create regular tablespace ts_orderline_037 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V2ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;

```

```

commit;

-- now creating TS for ts_orderline_038 of D1

drop tablespace ts_orderline_038;
create regular tablespace ts_orderline_038 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V3ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_039 of D1

drop tablespace ts_orderline_039;
create regular tablespace ts_orderline_039 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V4ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_040 of D1

drop tablespace ts_orderline_040;
create regular tablespace ts_orderline_040 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V5ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_041 of D1

drop tablespace ts_orderline_041;
create regular tablespace ts_orderline_041 pagesize 8K
managed by database
using
(
    device '/dev/rD1F09V1ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_042 of D1

drop tablespace ts_orderline_042;
create regular tablespace ts_orderline_042 pagesize 8K

```

```

managed by database
using
(
    device '/dev/rD1F09V2ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_043 of D1

drop tablespace ts_orderline_043;
create regular tablespace ts_orderline_043 pagesize 8K
managed by database
using
(
    device '/dev/rD1F09V3ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_044 of D1

drop tablespace ts_orderline_044;
create regular tablespace ts_orderline_044 pagesize 8K
managed by database
using
(
    device '/dev/rD1F09V4ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_045 of D1

drop tablespace ts_orderline_045;
create regular tablespace ts_orderline_045 pagesize 8K
managed by database
using
(
    device '/dev/rD1F09V5ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_046 of D1

drop tablespace ts_orderline_046;
create regular tablespace ts_orderline_046 pagesize 8K
managed by database
using
(
    device '/dev/rD1F10V1ORL' 7783936
)
extentsize 256

```

```

        prefetchsize 4096
        bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_047 of D1

drop tablespace ts_orderline_047;
create regular tablespace ts_orderline_047 pagesize 8K
managed by database
using
(
    device '/dev/rD1F10V2ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_048 of D1

drop tablespace ts_orderline_048;
create regular tablespace ts_orderline_048 pagesize 8K
managed by database
using
(
    device '/dev/rD1F10V3ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_049 of D1

drop tablespace ts_orderline_049;
create regular tablespace ts_orderline_049 pagesize 8K
managed by database
using
(
    device '/dev/rD1F10V4ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_050 of D1

drop tablespace ts_orderline_050;
create regular tablespace ts_orderline_050 pagesize 8K
managed by database
using
(
    device '/dev/rD1F10V5ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_051 of D1

```

```

drop tablespace ts_orderline_051;
create regular tablespace ts_orderline_051 pagesize 8K
managed by database
using
(
    device '/dev/rD1F11V1ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

connect reset;

```

ts/crts_stock.ddl

```

connect to tpcc;
-- now creating TS for ts_stock_001 of D1

drop tablespace ts_stock_001;
create regular tablespace ts_stock_001 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_002 of D1

drop tablespace ts_stock_002;
create regular tablespace ts_stock_002 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_003 of D1

drop tablespace ts_stock_003;
create regular tablespace ts_stock_003 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_004 of D1

drop tablespace ts_stock_004;
create regular tablespace ts_stock_004 pagesize 4K

```

```

managed by database
using
(
    device '/dev/rD1F01V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_005 of D1

drop tablespace ts_stock_005;
create regular tablespace ts_stock_005 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V5STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_006 of D1

drop tablespace ts_stock_006;
create regular tablespace ts_stock_006 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_007 of D1

drop tablespace ts_stock_007;
create regular tablespace ts_stock_007 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_008 of D1

drop tablespace ts_stock_008;
create regular tablespace ts_stock_008 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_009 of D1

```

```

drop tablespace ts_stock_009;
create regular tablespace ts_stock_009 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_010 of D1

drop tablespace ts_stock_010;
create regular tablespace ts_stock_010 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V5STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_011 of D1

drop tablespace ts_stock_011;
create regular tablespace ts_stock_011 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_012 of D1

drop tablespace ts_stock_012;
create regular tablespace ts_stock_012 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_013 of D1

drop tablespace ts_stock_013;
create regular tablespace ts_stock_013 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V3STK' 14001664
)
extentsize 256
prefetchsize 4096;

```

```

commit;

-- now creating TS for ts_stock_014 of D1

drop tablespace ts_stock_014;
create regular tablespace ts_stock_014 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_015 of D1

drop tablespace ts_stock_015;
create regular tablespace ts_stock_015 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V5STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_016 of D1

drop tablespace ts_stock_016;
create regular tablespace ts_stock_016 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_017 of D1

drop tablespace ts_stock_017;
create regular tablespace ts_stock_017 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_018 of D1

drop tablespace ts_stock_018;
create regular tablespace ts_stock_018 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V3STK' 14001664

```

```

)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_019 of D1

drop tablespace ts_stock_019;
create regular tablespace ts_stock_019 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_020 of D1

drop tablespace ts_stock_020;
create regular tablespace ts_stock_020 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V5STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_021 of D1

drop tablespace ts_stock_021;
create regular tablespace ts_stock_021 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_022 of D1

drop tablespace ts_stock_022;
create regular tablespace ts_stock_022 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_023 of D1

drop tablespace ts_stock_023;
create regular tablespace ts_stock_023 pagesize 4K
managed by database

```

```

using
(
    device '/dev/rD1F05V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_024 of D1

drop tablespace ts_stock_024;
create regular tablespace ts_stock_024 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_025 of D1

drop tablespace ts_stock_025;
create regular tablespace ts_stock_025 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V5STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_026 of D1

drop tablespace ts_stock_026;
create regular tablespace ts_stock_026 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_027 of D1

drop tablespace ts_stock_027;
create regular tablespace ts_stock_027 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_028 of D1

```

```

drop tablespace ts_stock_028;
create regular tablespace ts_stock_028 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_029 of D1

drop tablespace ts_stock_029;
create regular tablespace ts_stock_029 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_030 of D1

drop tablespace ts_stock_030;
create regular tablespace ts_stock_030 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V5STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_031 of D1

drop tablespace ts_stock_031;
create regular tablespace ts_stock_031 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_032 of D1

drop tablespace ts_stock_032;
create regular tablespace ts_stock_032 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_stock_033 of D1

drop tablespace ts_stock_033;
create regular tablespace ts_stock_033 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_034 of D1

drop tablespace ts_stock_034;
create regular tablespace ts_stock_034 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_035 of D1

drop tablespace ts_stock_035;
create regular tablespace ts_stock_035 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V5STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_036 of D1

drop tablespace ts_stock_036;
create regular tablespace ts_stock_036 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_037 of D1

drop tablespace ts_stock_037;
create regular tablespace ts_stock_037 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V2STK' 14001664
)

```

```

        extentsize 256
        prefetchsize 4096;
commit;

-- now creating TS for ts_stock_038 of D1

drop tablespace ts_stock_038;
create regular tablespace ts_stock_038 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_039 of D1

drop tablespace ts_stock_039;
create regular tablespace ts_stock_039 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_040 of D1

drop tablespace ts_stock_040;
create regular tablespace ts_stock_040 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V5STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_041 of D1

drop tablespace ts_stock_041;
create regular tablespace ts_stock_041 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_042 of D1

drop tablespace ts_stock_042;
create regular tablespace ts_stock_042 pagesize 4K
managed by database
using

```

```

(
    device '/dev/rD1F09V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_043 of D1

drop tablespace ts_stock_043;
create regular tablespace ts_stock_043 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_044 of D1

drop tablespace ts_stock_044;
create regular tablespace ts_stock_044 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_045 of D1

drop tablespace ts_stock_045;
create regular tablespace ts_stock_045 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V5STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_046 of D1

drop tablespace ts_stock_046;
create regular tablespace ts_stock_046 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_047 of D1

drop tablespace ts_stock_047;

```

```

create regular tablespace ts_stock_047 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_048 of D1

drop tablespace ts_stock_048;
create regular tablespace ts_stock_048 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_049 of D1

drop tablespace ts_stock_049;
create regular tablespace ts_stock_049 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_050 of D1

drop tablespace ts_stock_050;
create regular tablespace ts_stock_050 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V5STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_051 of D1

drop tablespace ts_stock_051;
create regular tablespace ts_stock_051 pagesize 4K
managed by database
using
(
    device '/dev/rD1F11V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

```

```
connect reset;
```

ts/crts ware.ddl

```
connect to tpcc;
```

```
-- now creating TS for ts_ware_001 of D1
```

```
drop tablespace ts_ware_001;  
create regular tablespace ts_ware_001 pagesize 4K  
managed by database  
using  
(  
    device '/dev/rD1F01V1WARE' 256  
)  
extentsize 32  
prefetchsize 4096;  
commit;
```

```
-- now creating TS for ts_ware_002 of D1
```

```
drop tablespace ts_ware_002;  
create regular tablespace ts_ware_002 pagesize 4K  
managed by database  
using  
(  
    device '/dev/rD1F01V2WARE' 256  
)  
extentsize 32  
prefetchsize 4096;  
commit;
```

```
-- now creating TS for ts_ware_003 of D1
```

```
drop tablespace ts_ware_003;  
create regular tablespace ts_ware_003 pagesize 4K  
managed by database  
using  
(  
    device '/dev/rD1F01V3WARE' 256  
)  
extentsize 32  
prefetchsize 4096;  
commit;
```

```
-- now creating TS for ts_ware_004 of D1
```

```
drop tablespace ts_ware_004;  
create regular tablespace ts_ware_004 pagesize 4K  
managed by database  
using  
(  
    device '/dev/rD1F01V4WARE' 256  
)  
extentsize 32  
prefetchsize 4096;  
commit;
```

```
-- now creating TS for ts_ware_005 of D1
```

```
drop tablespace ts_ware_005;
```

```
create regular tablespace ts_ware_005 pagesize 4K  
managed by database  
using  
(  
    device '/dev/rD1F01V5WARE' 256  
)  
extentsize 32  
prefetchsize 4096;  
commit;
```

```
-- now creating TS for ts_ware_006 of D1
```

```
drop tablespace ts_ware_006;  
create regular tablespace ts_ware_006 pagesize 4K  
managed by database  
using  
(  
    device '/dev/rD1F02V1WARE' 256  
)  
extentsize 32  
prefetchsize 4096;  
commit;
```

```
-- now creating TS for ts_ware_007 of D1
```

```
drop tablespace ts_ware_007;  
create regular tablespace ts_ware_007 pagesize 4K  
managed by database  
using  
(  
    device '/dev/rD1F02V2WARE' 256  
)  
extentsize 32  
prefetchsize 4096;  
commit;
```

```
-- now creating TS for ts_ware_008 of D1
```

```
drop tablespace ts_ware_008;  
create regular tablespace ts_ware_008 pagesize 4K  
managed by database  
using  
(  
    device '/dev/rD1F02V3WARE' 256  
)  
extentsize 32  
prefetchsize 4096;  
commit;
```

```
-- now creating TS for ts_ware_009 of D1
```

```
drop tablespace ts_ware_009;  
create regular tablespace ts_ware_009 pagesize 4K  
managed by database  
using  
(  
    device '/dev/rD1F02V4WARE' 256  
)  
extentsize 32  
prefetchsize 4096;  
commit;
```

```
-- now creating TS for ts_ware_010 of D1
```

```
drop tablespace ts_ware_010;  
create regular tablespace ts_ware_010 pagesize 4K  
managed by database  
using  
(  
    device '/dev/rD1F02V5WARE' 256  
)  
extentsize 32  
prefetchsize 4096;  
commit;
```

```
-- now creating TS for ts_ware_011 of D1
```

```
drop tablespace ts_ware_011;  
create regular tablespace ts_ware_011 pagesize 4K  
managed by database  
using  
(  
    device '/dev/rD1F03V1WARE' 256  
)  
extentsize 32  
prefetchsize 4096;  
commit;
```

```
-- now creating TS for ts_ware_012 of D1
```

```
drop tablespace ts_ware_012;  
create regular tablespace ts_ware_012 pagesize 4K  
managed by database  
using  
(  
    device '/dev/rD1F03V2WARE' 256  
)  
extentsize 32  
prefetchsize 4096;  
commit;
```

```
-- now creating TS for ts_ware_013 of D1
```

```
drop tablespace ts_ware_013;  
create regular tablespace ts_ware_013 pagesize 4K  
managed by database  
using  
(  
    device '/dev/rD1F03V3WARE' 256  
)  
extentsize 32  
prefetchsize 4096;  
commit;
```

```
-- now creating TS for ts_ware_014 of D1
```

```
drop tablespace ts_ware_014;  
create regular tablespace ts_ware_014 pagesize 4K  
managed by database  
using  
(  
    device '/dev/rD1F03V4WARE' 256  
)  
extentsize 32
```

```

        prefetchsize 4096;
commit;

-- now creating TS for ts_ware_015 of D1

drop tablespace ts_ware_015;
create regular tablespace ts_ware_015 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V5WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_016 of D1

drop tablespace ts_ware_016;
create regular tablespace ts_ware_016 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V1WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_017 of D1

drop tablespace ts_ware_017;
create regular tablespace ts_ware_017 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V2WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_018 of D1

drop tablespace ts_ware_018;
create regular tablespace ts_ware_018 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V3WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_019 of D1

drop tablespace ts_ware_019;
create regular tablespace ts_ware_019 pagesize 4K
managed by database
using
(

```

```

        device '/dev/rD1F04V4WARE' 256
    )
    extentsize 32
    prefetchsize 4096;
commit;

-- now creating TS for ts_ware_020 of D1

drop tablespace ts_ware_020;
create regular tablespace ts_ware_020 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V5WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_021 of D1

drop tablespace ts_ware_021;
create regular tablespace ts_ware_021 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V1WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_022 of D1

drop tablespace ts_ware_022;
create regular tablespace ts_ware_022 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V2WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_023 of D1

drop tablespace ts_ware_023;
create regular tablespace ts_ware_023 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V3WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_024 of D1

drop tablespace ts_ware_024;
create regular tablespace ts_ware_024 pagesize 4K

```

```

managed by database
using
(
    device '/dev/rD1F05V4WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_025 of D1

drop tablespace ts_ware_025;
create regular tablespace ts_ware_025 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V5WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_026 of D1

drop tablespace ts_ware_026;
create regular tablespace ts_ware_026 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V1WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_027 of D1

drop tablespace ts_ware_027;
create regular tablespace ts_ware_027 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V2WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_028 of D1

drop tablespace ts_ware_028;
create regular tablespace ts_ware_028 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V3WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_029 of D1

```

```

drop tablespace ts_ware_029;
create regular tablespace ts_ware_029 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V4WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_030 of D1

drop tablespace ts_ware_030;
create regular tablespace ts_ware_030 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V5WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_031 of D1

drop tablespace ts_ware_031;
create regular tablespace ts_ware_031 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V1WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_032 of D1

drop tablespace ts_ware_032;
create regular tablespace ts_ware_032 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V2WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_033 of D1

drop tablespace ts_ware_033;
create regular tablespace ts_ware_033 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V3WARE' 256
)
extentsize 32
prefetchsize 4096;

```

```

commit;

-- now creating TS for ts_ware_034 of D1

drop tablespace ts_ware_034;
create regular tablespace ts_ware_034 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V4WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_035 of D1

drop tablespace ts_ware_035;
create regular tablespace ts_ware_035 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V5WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_036 of D1

drop tablespace ts_ware_036;
create regular tablespace ts_ware_036 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V1WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_037 of D1

drop tablespace ts_ware_037;
create regular tablespace ts_ware_037 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V2WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_038 of D1

drop tablespace ts_ware_038;
create regular tablespace ts_ware_038 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V3WARE' 256

```

```

)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_039 of D1

drop tablespace ts_ware_039;
create regular tablespace ts_ware_039 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V4WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_040 of D1

drop tablespace ts_ware_040;
create regular tablespace ts_ware_040 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V5WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_041 of D1

drop tablespace ts_ware_041;
create regular tablespace ts_ware_041 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V1WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_042 of D1

drop tablespace ts_ware_042;
create regular tablespace ts_ware_042 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V2WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_043 of D1

drop tablespace ts_ware_043;
create regular tablespace ts_ware_043 pagesize 4K
managed by database

```



```

using
(
    device '/dev/rD1F09V3WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_044 of D1

drop tablespace ts_ware_044;
create regular tablespace ts_ware_044 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V4WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_045 of D1

drop tablespace ts_ware_045;
create regular tablespace ts_ware_045 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V5WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_046 of D1

drop tablespace ts_ware_046;
create regular tablespace ts_ware_046 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V1WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_047 of D1

drop tablespace ts_ware_047;
create regular tablespace ts_ware_047 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V2WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_048 of D1

```

```

drop tablespace ts_ware_048;
create regular tablespace ts_ware_048 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V3WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_049 of D1

drop tablespace ts_ware_049;
create regular tablespace ts_ware_049 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V4WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_050 of D1

drop tablespace ts_ware_050;
create regular tablespace ts_ware_050 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V5WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_051 of D1

drop tablespace ts_ware_051;
create regular tablespace ts_ware_051 pagesize 4K
managed by database
using
(
    device '/dev/rD1F11V1WARE' 256
)
extentsize 32
prefetchsize 4096;
commit;

connect reset;

```

C.2 Data Generation Code

Makefile.config

```

#####
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp.
## 1996 - 2006
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication
or
## disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
#####
#####

#
# Makefile.config - AIX 64-bit
#

# Make Configuration
MAKE=make

# Compiler Configuration.
# CFLAGS_DEBUG may be set to "-g", "-DDEBUGIT" "-g -
DDEBUGIT" or left blank
CC=xlC
CFLAGS_OS=-qflag=:i -qlanglvl=ansi -qcpluscmt -DSQLUNIX -
DSQLAIX -q64 -O3 -D_LARGE_FILES
CFLAGS_OUT=-o
CFLAGS_DEBUG=

# Linker Configuration
LD_EXEC=xlC
LD_STORP=xlC
LDFLAGS_EXEC=-q64
LDFLAGS_SHLIB=qmkshrobj
LDFLAGS_STORP=$(LDFLAGS_SHLIB) -bE:$@.exp -lc -b64
LDFLAGS_LIB=-L$(TPCC_SQLLIB)/lib -ldb2
LDFLAGS_OUT=-o

# Library Configuration
AR=ar
ARFLAGS=-r -v -X64
ARFLAGS_LIB=
ARFLAGS_OUT=

# OS Commands
ERASE=rm -f
ERASEDIR=$(ERASE) -R
MOVE=mv
COPY=cp

# OS File Extensions & Path Separators

```

```
OBJEXT=.o
LIBEXT=.a
SHLIBEXT=.a
BINEXT=
SLASH=/
CMDSEP=;
```

Src.Common/Makefile

```
#####
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp.
1996 - 2006
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication
or
## disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
#####
#####
#
# Makefile - Makefile for Src.Common
#

include $(TPCC_ROOT)/Makefile.config

#
#####
#####
# Preprocessor, Compiler and Linker Flags
#
#####
#####

BND_OPTS = GRANT PUBLIC \
            MESSAGES $.bnd.msg
PRP_OPTS = BINDFILE \
            OPTLEVEL 1 \
            ISOLATION RR \
            MESSAGES $.prep.msg \
            LEVEL $(TPCC_VERSION) \
            NOLINEMACRO

INCLUDE = -I$(TPCC_SQLLIB)/include -
I$(TPCC_ROOT)/include

CFLAGS = $(CFLAGS_OS) $(CFLAGS_DEBUG) $(INCLUDE) \
         -DSQLA_NOLINES -D$(DB2EDITION) -
D$(DB2VERSION) \
         -D$(TPCC_SPTYPE)

UTIL_OBJ = tpcmisc$(OBJEXT) tpcdbg$(OBJEXT)
tpccctx$(OBJEXT)
```

```
#
#####
#####
# User Targets
#
#####
#####

all: connect $(UTIL_OBJ) disconnect

clean:
- $(ERASE) *$(OBJEXT) *.bnd *.msg tpcctx.c

#
#####
#####
# Helper Targets
#
#####
#####

connect:
- db2 connect to $(TPCC_DBNAME)

disconnect:
- db2 connect reset
- db2 terminate

rebind: connect
db2 bind tpcctx.bnd $(BND_OPTS)

#
#####
#####
# Build Rules
#
#####
#####

.SUFFIXES:
.SUFFIXES: $(OBJEXT) .c .sqc

.sqc.c:
@echo "Prepping $.sqc"
-db2 prep $.sqc $(PRP_OPTS)
@echo "Binding $.bnd"
db2 bind $.bnd $(BND_OPTS)

#
#####
#####
# Dependencies
#
#####
#####

# Source
tpccdbg$(OBJEXT): tpcdbg.c
tpccctx$(OBJEXT): tpcctx.c
tpcmisc$(OBJEXT): tpcmisc.c

# Headers
```

tpccdbg.c: \$(TPCC_ROOT)/include/db2tpcc.h

Src.Common/tpcmisc.c

```
/*
****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp.
1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
*****
****/

/*
* tpcmisc.c - Miscellaneous routines
*/

#include <stdlib.h>
#include <sys/types.h>
#include <sys/time.h>

double current_time_ms(void);
double current_time(void);

/* Current time in SECONDS, precision SECONDS */
double current_time(void)
{
/* use time() to get seconds */
return(time(NULL));
}

/* Current time in SECONDS, precision MILLISECONDS */
double current_time_ms(void)
{
/* gettimeofday() returns seconds and microseconds */
/* convert to fractional seconds */
struct timeval t;
gettimeofday(&t,NULL);
return (t.tv_sec + (double)t.tv_usec/(1000*1000));
}

dbgen/Makefile

#####
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp.
1996 - 2006
```



```

ware_start = 1;
ware_end = WAREHOUSES;

/* ***** */
/* Process Command Line Arguments */
/* ***** */

/* Valid Command Line Options
* -----
* Table Option:      -t <table>      (-t 3 for warehouse)
* Output Column Delimiter: -d <char>  (-d ' ', -d '|', etc)
* Output to File:    -f[n] <file>    (-f customer.dat)
* Output to Pipe:    -p[n] <pipename> (-p tpccpipe.000)
* Warehouse Range:  -r <start> <end> (-r 1 100)
* Scaling Report:    -s
* Quiet Mode:        -q
*
* The -f[n] and/or -p[n] options are required.
* The -t, -d, -r, -s and -q options are optional.
*
* If -d is omitted, the vertical bar (pipe) symbol (|) will be
used.
* If -r is omitted, the range [1..WAREHOUSES] will be used.
*
* Due to the TPC-C spec requiring that orders and orderline
be
* generated at the same time, there is an extension to the -f
and -p
* options to specify one of the two output streams for each
argument.
*
* -f1 orders.dat -f2 orderline.dat will output to two files
* -f1 orders.dat -p2 tpccpipe.000 will output to a file and a
pipe
*
* -f1/-p1 specifies the destination for the orders table
* -f2/-p2 specifies the destination for the orderline table
*
*/

/* Read Arguments */
for (i=1; i<argc; i++)
{
    if (strcmp(argv[i], "-t") == 0) {
        option = atoi(argv[i+1]);
        i++;
    } else if (strcmp(argv[i], "-r") == 0) {
        ware_start = atoi(argv[i+1]);
        ware_end = atoi(argv[i+2]);
        i += 2;
    } else if (strcmp(argv[i], "-d") == 0) {
        delim = argv[i+1];
        i++;
    } else if ((strcmp(argv[i], "-f") == 0) ||
                (strcmp(argv[i], "-f1") == 0)) {
        outtype1 = IOH_FILE;
        outname1 = argv[i+1];
        i++;
    } else if (strcmp(argv[i], "-f2") == 0) {
        outtype2 = IOH_FILE;
        outname2 = argv[i+1];
        i++;
    }
}

```

```

    } else if ((strcmp(argv[i], "-p") == 0) ||
                (strcmp(argv[i], "-p1") == 0)) {
        outtype1 = IOH_PIPE;
        outname1 = argv[i+1];
        i++;
    } else if (strcmp(argv[i], "-p2") == 0) {
        outtype2 = IOH_PIPE;
        outname2 = argv[i+1];
        i++;
    } else if (strcmp(argv[i], "-s") == 0) {
        ScalingReport();
        exit(0);
    } else if (strcmp(argv[i], "-q") == 0) {
        quiet_mode = 1;
    } else {
        fprintf(stderr, "gendata: Don't understand argument:
%s\n", argv[i]);
        exit(-1);
    }
}

/* ***** */
/* Validate Command Line Arguments */
/* ***** */

/* Validate Table Argument */
if (option < 3 || option > 11 || option == 10)
{
    fprintf(stderr, "gendata: Invalid table selected: %d\n", option);
    exit(-1);
}

/* Validate Delimiter Argument */
if (delim == NULL) {
    // default delimiter is used for IMPORT & LOAD, no changes
necessary
    using_rctload = 0;
} else if (strlen(delim) == 1 && !isalnum(delim[0]) &&
           delim[0] != '.' && delim[0] != '%')
{
    // user-supplied delimiter used for rctload
    InitFormatStrings(delim[0]);
    using_rctload = 1;
} else {
    fprintf(stderr, "gendata: Invalid delimiter specified:
%s\n", delim);
    exit(-1);
}

/* Validate File/Pipe Arguments */
if (option != 9 && outtype1 > 0 && outtype2 > 0)
{
    fprintf(stderr, "gendata: Specifying two output file/pipes
allowed only when generating\norders/orderline.\n");
    exit(-1);
}
if (option == 9 && ((outtype1 == 0) || (outtype2 == 0)))
{
    fprintf(stderr, "gendata: Must specify two output file/pipes
when generating orders/orderline.\n");
    exit(-1);
}
}

```

```

    if (outtype1 == 0 || outname1 == NULL || strcmp(outname1, "")
== 0)
    {
        fprintf(stderr, "gendata: Invalid 1st output file/pipe
specified.\n");
        exit(-1);
    }
    if (option == 9 && (outtype2 == 0 || outname2 == NULL ||
strcmp(outname2, "") == 0))
    {
        fprintf(stderr, "gendata: Invalid 2nd output file/pipe
specified.\n");
        exit(-1);
    }
    /* Ensure O/OL flat files are opened in append mode. This is
required */
    /* because we generate O/OL concurrently. See comments in
genload.pl */
    /* for further details on why this is necessary. */
    if (option == 9)
    {
        if (outtype1 == IOH_FILE) outtype1 = IOH_FILE_APPEND;
        if (outtype2 == IOH_FILE) outtype2 = IOH_FILE_APPEND;
    }

    /* Validate Range Arguments */
    if (ware_start <= 0 || ware_start > WAREHOUSES) {
        fprintf(stderr, "gendata: Invalid range starting value:
%d\n", ware_start);
        exit(-1);
    }
    if (ware_end <= 0 || ware_end > WAREHOUSES || ware_end
< ware_start) {
        fprintf(stderr, "gendata: Invalid range ending value:
%d\n", ware_end);
        exit(-1);
    }

    /* ***** */
    /* Generate Data */
    /* ***** */
    switch (option) {
        case 3: /* WAREHOUSE */
            gen_ware_tbl();
            break;
        case 4: /* DISTRICT */
            gen_dist_tbl();
            break;
        case 5: /* ITEM */
            gen_item_tbl();
            break;
        case 6: /* STOCK */
            gen_stock_tbl();
            break;
        case 7: /* CUSTOMER */
            gen_cust_tbl();
            break;
        case 8: /* HISTORY */
            gen_hist_tbl();
            break;
        case 9: /* ORDERS + ORDER_LINE */
            gen_ordr_tbl();

```

```

    break;
case 11: /* NEW_ORDER */
    gen_nu_ord_tbl();
    break;
case 2:
case 10:
default:
    fprintf(stderr, "Error: invalid option = %d \n", (option));
    break;
}
return 0;
}

/*-----*/
/* generate item table */
/*-----*/

void gen_item_tbl( void )
{
    sqlint32 item_num = 0 ;
    sqlint32 item_im_id ;
    char item_name[25];
    sqlint32 item_price ;
    char item_data[51];

    IOH_NUM numBytes;
    ioHandle hnd;
    char Buffer[1024];

    initialize_random(13,42);
    timestamp1 = current_time();

    rc = GenericOpen(&hnd, outtype1, outname1);
    if (rc != 0) { goto item_done; }

    for(item_num = 1; item_num <= ITEMS; item_num++)
    {
        /* create image id field */
        item_im_id = rand_integer( 1, 10000 );
        /* create name field */
        create_random_a_string( item_name, 14, 24);
        /* create price field */
        item_price = rand_integer( 100, 10000 );
        /* create ORIGINAL field */
        create_a_string_with_original( item_data, 26, 50, 10 );

        numBytes = sprintf(Buffer, fmtItem,
            item_name,
            item_price,
            item_data,
            item_im_id,
            item_num);

        rc = GenericWrite(&hnd, Buffer, numBytes);
        if (rc != 0) { goto item_done; }

    } /* end for... */

    rc = GenericClose(&hnd);

item_done:

```

```

    timestamp2 = current_time();
    elapse = timestamp2 - timestamp1;
    if (rc == 0) {
        fprintf(stdout, "\nITEM table generated in %8.2f
seconds.\n\n", elapse);
        fflush(stdout);
    } else {
        fprintf(stderr, "\nITEM table FAILED at (I %d) after %8.2f
seconds.\n\n", item_num, elapse);
        fflush(stderr);
    }
}

/*-----*/
/* generate stock table */
/*-----*/

void gen_stock_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 stock_num = 0 ;
    sqlint32 stock_quant;
    sqlint32 s_ytd;
    sqlint32 s_order_cnt, s_remote_cnt;
    char stock_dist_01[25];
    char stock_dist_02[25];
    char stock_dist_03[25];
    char stock_dist_04[25];
    char stock_dist_05[25];
    char stock_dist_06[25];
    char stock_dist_07[25];
    char stock_dist_08[25];
    char stock_dist_09[25];
    char stock_dist_10[25];
    char stock_data[51];

    IOH_NUM numBytes;
    ioHandle hnd;
    char Buffer[1024];

    initialize_random(7,11);
    timestamp1 = current_time();

    rc = GenericOpen(&hnd, outtype1, outname1);
    if (rc != 0) { goto stock_done; }

    for (stock_num = 1; stock_num <=
STOCK_PER_WAREHOUSE; stock_num++)
    {
        if (!quiet_mode && (stock_num%500 == 0))
        {
            fprintf(stdout, "STOCK for Item #%d\n", stock_num);
            fflush(stdout);
        }
        for (ware_num = ware_start; ware_num <= ware_end;
ware_num++)
        {
            stock_quant = rand_integer( 10, 100 );
            create_random_a_string( stock_dist_01, 24, 24);
            create_random_a_string( stock_dist_02, 24, 24);
            create_random_a_string( stock_dist_03, 24, 24);
            create_random_a_string( stock_dist_04, 24, 24);
            create_random_a_string( stock_dist_05, 24, 24);

```

```

            create_random_a_string( stock_dist_06, 24, 24);
            create_random_a_string( stock_dist_07, 24, 24);
            create_random_a_string( stock_dist_08, 24, 24);
            create_random_a_string( stock_dist_09, 24, 24);
            create_random_a_string( stock_dist_10, 24, 24);

        /* create ORIGINAL field */
        create_a_string_with_original( stock_data, 26, 50, 10 );
        s_ytd = s_order_cnt = s_remote_cnt = 0;

        numBytes = sprintf(Buffer, fmtStock,
            s_remote_cnt,
            stock_quant,
            s_order_cnt,
            s_ytd,
            stock_data,
            stock_dist_01,
            stock_dist_02,
            stock_dist_03,
            stock_dist_04,
            stock_dist_05,
            stock_dist_06,
            stock_dist_07,
            stock_dist_08,
            stock_dist_09,
            stock_dist_10,
            stock_num,
            ware_num);

        rc = GenericWrite(&hnd, Buffer, numBytes);
        if (rc != 0) { goto stock_done; }

    } /* end for... */
} /* end for... */

rc = GenericClose(&hnd);

stock_done:

    timestamp2 = current_time();
    elapse = timestamp2 - timestamp1;
    if (rc == 0) {
        fprintf(stdout, "\nSTOCK table generated in %8.2f
seconds.\n\n", elapse);
        fflush(stdout);
    } else {
        fprintf(stderr, "\nSTOCK table FAILED at (S %d W %d) after
%8.2f seconds.\n\n", stock_num, ware_num, elapse);
        fflush(stderr);
    }
}

/*-----*/
/* generate warehouse table */
/*-----*/

void gen_ware_tbl( void )
{
    sqlint32 ware_num = 0 ;
    char ware_name[11];
    char ware_street_1[21];
    char ware_street_2[21];
    char ware_city[21];

```

```

char ware_state[3] ;
char ware_zip[10] ;
sqlint32 ware_tax ;
sqlint64 ware_YTD ;

IOH_NUM numBytes;
ioHandle hnd;
char Buffer[1024];

initialize_random(23,111);
timestamp1 = current_time();

rc = GenericOpen(&hnd, outtype1, outname1);
if (rc != 0) { goto ware_done; }

for (ware_num = ware_start; ware_num <= ware_end;
ware_num++)
{
    if (!quiet_mode && ((ware_num % 500) == 0))
    {
        //@dxxxxxmt
        fprintf(stdout, "Warehouse #%"d\n", ware_num);
        fflush(stdout);
    }

    create_random_a_string( ware_name, 6,10); /* create
name */
    create_random_a_string( ware_street_1, 10,20); /* create
street 1 */
    create_random_a_string( ware_street_2, 10,20); /* create
street 2 */
    create_random_a_string( ware_city, 10,20); /* create city
*/
    create_random_a_string( ware_state, 2,2); /* create
state */
    create_random_n_string( ware_zip, 4,4); /* create zip */
    strcat(ware_zip, "11111");

    ware_tax = rand_integer(0, 2000);
    ware_YTD = 30000000;

    numBytes = sprintf(Buffer, fmtWare,
        ware_name,
        ware_street_1,
        ware_street_2,
        ware_city,
        ware_state,
        ware_zip,
        ware_tax,
        ware_YTD,
        ware_num);

    rc = GenericWrite(&hnd, Buffer, numBytes);
    if (rc != 0) { goto ware_done; }

} /* end for */

rc = GenericClose(&hnd);

ware_done:

timestamp2 = current_time();
elapsed = timestamp2 - timestamp1;

```

```

if (rc == 0) {
    fprintf(stdout, "\nWAREHOUSE table generated in %8.2f
seconds.\n\n", elapsed);
    fflush(stdout);
} else {
    fprintf(stderr, "\nWAREHOUSE table FAILED at (W %d) after
%8.2f seconds.\n\n", ware_num, elapsed);
    fflush(stderr);
}
}

/*-----*/
/* generate dist table */
/*-----*/

void gen_dist_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    char dist_name[11];
    char dist_street_1[21];
    char dist_street_2[21];
    char dist_city[21];
    char dist_state[3];
    char dist_zip[10];
    sqlint32 dist_tax;
    sqlint32 next_o_id;
    sqlint64 dist_YTD;

    IOH_NUM numBytes;
    ioHandle hnd;
    char Buffer[1024];

    next_o_id = CUSTOMERS_PER_DISTRICT + 1;
    initialize_random(44,73);
    timestamp1 = current_time();

    rc = GenericOpen(&hnd, outtype1, outname1);
    if (rc != 0) { goto dist_done; }

    for (ware_num = ware_start; ware_num <= ware_end;
ware_num++)
    {
        if (!quiet_mode) {
            fprintf(stdout, "DISTRICT for Warehouse #%"d\n",
ware_num);
            fflush(stdout);
        }
        for (dist_num = 1; dist_num <=
DISTRICTS_PER_WAREHOUSE; dist_num++)
        {
            create_random_a_string( dist_name, 6,10); /* create
name */
            create_random_a_string( dist_street_1, 10,20); /* create
street 1 */
            create_random_a_string( dist_street_2, 10,20); /* create
street 2 */
            create_random_a_string( dist_city, 10,20); /* create city
*/
            create_random_a_string( dist_state, 2,2); /* create
state */
            create_random_n_string( dist_zip, 4,4); /* create zip
*/

```

```

        strcat(dist_zip, "11111");
        dist_tax = rand_integer(0, 2000);
        dist_YTD = 3000000;

        numBytes = sprintf(Buffer, fmtDist,
            next_o_id,
            dist_tax,
            dist_YTD,
            dist_name,
            dist_street_1,
            dist_street_2,
            dist_city,
            dist_state,
            dist_zip,
            dist_num,
            ware_num);

        rc = GenericWrite(&hnd, Buffer, numBytes);
        if (rc != 0) { goto dist_done; }

    } /* end for... */
} /* end for... */

rc = GenericClose(&hnd);

dist_done:

timestamp2 = current_time();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nDISTRICT table generated in %8.2f
seconds.\n\n", elapsed);
    fflush(stdout);
} else {
    fprintf(stderr, "\nDISTRICT table FAILED at (W %d D %d)
after %8.2f seconds.\n\n", ware_num, dist_num, elapsed);
    fflush(stderr);
}
}

/*-----*/
/* generate customer table */
/*-----*/

void gen_cust_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    sqlint32 cust_num = 0 ;
    char cust_last[17];
    char cust_middle[3];
    char cust_first[17];
    char cust_street_1[21];
    char cust_street_2[21];
    char cust_city[21];
    char cust_state[3];
    char cust_zip[10];
    char cust_phone[17];
    char cust_credit[3];
    char cust_data[501];
    sqlint32 cust_discount;
    sqlint64 currmtstp;
    sqlint64 cust_balance;

```

```

sqlint64 cust_YTD_payment;
sqlint64 cust_credit_lim;

IOH_NUM numBytes;
ioHandle hnd;
char Buffer[1024];
int len, pos;

initialize_random(10,64);
timestamp1 = current_time();

rc = GenericOpen(&hnd, outtype1, outname1);
if (rc != 0) { goto cust_done; }

strcpy(cust_middle, "OE");
currtmstp = time(NULL);

for (cust_num = 1; cust_num <=
CUSTOMERS_PER_DISTRICT; cust_num++)
{
    if (!quiet_mode) {
        fprintf(stdout, "CUSTOMER #:%d:\n", cust_num);
        fflush(stdout);
    }

    for (ware_num = ware_start; ware_num <= ware_end;
ware_num++)
    {
        for (dist_num = 1; dist_num <=
DISTRICTS_PER_WAREHOUSE; dist_num++)
        {
            if (cust_num <= 1000) /* create last name
*/
                create_random_last_name( cust_last, cust_num);
            else /* create last name */
                create_random_last_name( cust_last, 0);
            create_random_a_string( cust_first, 8,16); /* create
first name */
            create_random_a_string( cust_street_1, 10,20); /*
create street 1 */
            create_random_a_string( cust_street_2, 10,20); /*
create street 2 */
            create_random_a_string( cust_city, 10,20); /* create
city */
            create_random_a_string( cust_state, 2,2); /* create
state */
            create_random_n_string( cust_zip, 4,4); /* create
zip */
            strcat(cust_zip, "11111");

            /* create phone number */
            create_random_n_string( cust_phone, 16,16);
            if ( rand_integer( 1, 100 ) <= 10 )
                strcpy( cust_credit, "BC" );
            else
                strcpy( cust_credit, "GC" );

            /* create discount rate */
            cust_discount = rand_integer(0, 5000);

            /* create customer data */
            create_random_a_string(cust_data, 300, 500);

```

```

/* pad customer data (only for non-rcload) */
if (using_rcload == 0) {
    for (pos=strlen(cust_data); pos<500; pos++)
        cust_data[pos] = ' ';
        cust_data[500] = '\0';
    }

cust_credit_lim = 5000000;
cust_balance = -1000;
cust_YTD_payment = 1000;

numBytes = sprintf(Buffer, fmtCust,
    cust_num,
    cust_state,
    cust_zip,
    cust_phone,
    currtmstp,
    cust_credit_lim,
    cust_middle,
    cust_credit,
    cust_discount,
    cust_data,
    cust_last,
    cust_first,
    cust_street_1,
    cust_street_2,
    cust_city,
    dist_num,
    ware_num,
    0,
    cust_balance,
    cust_YTD_payment,
    1);

rc = GenericWrite(&hnd, Buffer, numBytes);
if (rc != 0) { goto cust_done; }

} /* end for district... */
} /* end for warehouse... */
} /* end for customer... */

rc = GenericClose(&hnd);

cust_done:

timestamp2 = current_time();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nCUSTOMER table generated in %8.2f
seconds.\n\n", elapsed);
    fflush(stdout);
} else {
    fprintf(stderr, "\nCUSTOMER table FAILED at (W %d D %d
C %d) after %8.2f seconds.\n\n", ware_num, dist_num,
    cust_num, elapsed);
    fflush(stderr);
}
}

}

/*-----*/
/* generate hist table */

```

```

/*-----*/
void gen_hist_tbl( void )
{
    sqlint32 ware_num = 0;
    sqlint32 dist_num = 0;
    sqlint32 cust_num = 0;
    char hist_data[25];
    sqlint64 currtmstp;

    IOH_NUM numBytes;
    ioHandle hnd;
    char Buffer[1024];

    initialize_random(15,63);
    timestamp1 = current_time();

    rc = GenericOpen(&hnd, outtype1, outname1);
    if (rc != 0) { goto hist_done; }

    currtmstp = time(NULL);

    for (ware_num = ware_start; ware_num <= ware_end;
ware_num++)
    {
        if (!quiet_mode) {
            fprintf(stdout, "HISTORY for Warehouse #:%d:\n",
ware_num);
            fflush(stdout);
        }
        for (dist_num = 1; dist_num <=
DISTRICTS_PER_WAREHOUSE; dist_num++)
        {
            for (cust_num = 1; cust_num <=
CUSTOMERS_PER_DISTRICT; cust_num++)
            {
                /* create history data */
                create_random_a_string( hist_data, 12,24);

                numBytes = sprintf(Buffer, fmtHist,
                    cust_num,
                    dist_num,
                    ware_num,
                    dist_num,
                    ware_num,
                    currtmstp,
                    1000,
                    hist_data);

                rc = GenericWrite(&hnd, Buffer, numBytes);
                if (rc != 0) { goto hist_done; }

            } /* end for customer... */
        } /* end for district... */
    } /* end for warehouse... */

    rc = GenericClose(&hnd);

hist_done:

    timestamp2 = current_time();
    elapsed = timestamp2 - timestamp1;
    if (rc == 0) {

```

```

    fprintf(stdout, "\nHISTORY table generated in %8.2f
seconds.\n\n", elapse);
    fflush(stdout);
} else {
    fprintf(stderr, "\nHISTORY table FAILED at (W %d D %d C
%d) after %8.2f seconds.\n\n", ware_num, dist_num, cust_num,
elapse);
    fflush(stderr);
}
}
}
/*-----*/
/* generate nu_ord table */
/*-----*/
void gen_nu_ord_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    sqlint32 nu_ord_id = 0 ;
    int nu_ord_hi ;

    IOH_NUM numBytes;
    ioHandle hnd;
    char Buffer[1024];

    /* compute maximum and minimum
order numbers for this
district */
    nu_ord_hi = CUSTOMERS_PER_DISTRICT -
NU_ORDERS_PER_DISTRICT + 1;
    if (nu_ord_hi < 0) {
        nu_ord_hi = CUSTOMERS_PER_DISTRICT -
(CUSTOMERS_PER_DISTRICT / 3) + 1;
        fprintf(stderr, "\n**** WARNING ****
NU_ORDERS_PER_DISTRICT is >
CUSTOMERS_PER_DISTRICT\n");
        fprintf(stderr, "        Check the values in file lval.h\n");
        fprintf(stderr, "        Loading New-Order with 1/3 of
CUSTOMERS_PER_DISTRICT\n");
    }
    initialize_random(99,37);
    timestamp1 = current_time();

    rc = GenericOpen(&hnd, outtype1, outname1);
    if (rc != 0) { goto neword_done; }

    for (nu_ord_id = nu_ord_hi;
        nu_ord_id <= CUSTOMERS_PER_DISTRICT;
        nu_ord_id++)
    {
        if (!quiet_mode) {
            fprintf(stdout, "NEW_ORDER for Customer #%d:\n",
nu_ord_id);
            fflush(stdout);
        }
        for (ware_num = ware_start; ware_num <= ware_end;
ware_num++)
        {
            for (dist_num = 1; dist_num <=
DISTRICTS_PER_WAREHOUSE; dist_num++)
            {
                numBytes = sprintf(Buffer, fmtNewOrd,

```

```

nu_ord_id,
dist_num,
ware_num);

        rc = GenericWrite(&hnd, Buffer, numBytes);
        if (rc != 0) { goto neword_done; }

        /* end for... */
        /* end for... */
        /* end for... */
    }

    rc = GenericClose(&hnd);

neword_done:

    timestamp2 = current_time();
    elapse = timestamp2 - timestamp1;
    if (rc == 0) {
        fprintf(stdout, "\nNEW_ORDER table generated in %8.2f
seconds.\n\n", elapse);
        fflush(stdout);
    } else {
        fprintf(stderr, "\nNEW_ORDER table FAILED at (W %d D %d
O %d) after %8.2f seconds.\n\n", ware_num, dist_num,
nu_ord_id, elapse);
        fflush(stderr);
    }
}
/*-----*/
/* generate order and order_line tables */
/*-----*/
void gen_ordr_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    sqlint32 cust_num = 0 ;
    sqlint32 ord_num = 0 ;
    sqlint32 ordr_carrier_id;
    sqlint32 ordr_ol_cnt;
    sqlint32 oline_ol_num;
    sqlint32 oline_item_num;

    sqlint32 oline_amount;
    char oline_dist_info[25];
    sqlint64 nulltmstmp = 0;
    sqlint64 currtmstmp;

    IOH_NUM numBytes;
    ioHandle hnd1, hnd2;
    char Buffer[1024];

    oline_dist_info[24] = '\0';
    initialize_random(42,13);
    timestamp1 = current_time();

    rc1 = GenericOpen(&hnd1, outtype1, outname1);
    if (rc1 != 0) { goto ool_done; }
    rc2 = GenericOpen(&hnd2, outtype2, outname2);
    if (rc2 != 0) { goto ool_done; }

    currtmstmp = time(NULL);

```

```

    for (ware_num = ware_start; ware_num <= ware_end;
ware_num++)
    {
        if (!quiet_mode) {
            fprintf(stdout, "ORDERS & ORDER_LINE for Warehouse
#%d\n", ware_num);
            fflush(stdout);
        }
        for (dist_num = 1; dist_num <=
DISTRICTS_PER_WAREHOUSE; dist_num++)
        {
            if (!quiet_mode) {
                fprintf(stdout, "District #%d\t", dist_num);
                fflush(stdout);
            }

            seed_1_3000();

            for (ord_num = 1; ord_num <=
CUSTOMERS_PER_DISTRICT; ord_num++)
            {
                if (ord_num < 2101)
                    ordr_carrier_id = rand_integer( 1, 10 ) ;
                else
                    ordr_carrier_id = 0;

                cust_num = random_1_3000();
                ordr_ol_cnt =
rand_integer(MIN_OL_PER_ORDER, MAX_OL_PER_ORDER);

                numBytes = sprintf(Buffer, fmtOrdr,
                    cust_num,
                    currtmstmp,
                    ordr_carrier_id,
                    ordr_ol_cnt,
                    1,
                    ord_num,
                    ware_num,
                    dist_num);

                rc1 = GenericWrite(&hnd1, Buffer, numBytes);
                if (rc1 != 0) { goto ool_done; }

                for ( oline_ol_num = 1; oline_ol_num <= ordr_ol_cnt;
oline_ol_num++ )
                {
                    oline_item_num = rand_integer(1, ITEMS) ;
                    create_random_a_string( oline_dist_info, 24, 24) ;

                    numBytes = sprintf(Buffer, fmtOLine,
                        ((ord_num < 2101) ? currtmstmp :
nulltmstmp),
                        ((ord_num < 2101) ? 0 :
rand_integer(1,999999)),
                        oline_item_num,
                        ware_num,
                        5,
                        oline_dist_info,
                        ord_num,
                        dist_num,
                        ware_num,

```



```

        oline_ol_num);

rc2 = GenericWrite(&hnd2, Buffer, numBytes);
if (rc2 != 0) { goto ool_done; }

    } /* for order_line */
    } /* for order */
    } /* for dist */
} /* for ware */

rc1 = GenericClose(&hnd2);
rc2 = GenericClose(&hnd1);

ool_done:

timestamp2 = current_time();
elapsed = timestamp2 - timestamp1;
if (rc1 == 0 && rc2 == 0) {
    fprintf(stdout, "\nORDERS & ORDER_LINE table(s)
generated in %8.2f seconds.\n\n", elapsed);
    fflush(stdout);
} else {
    fprintf(stderr, "\nORDERS & ORDER_LINE table(s) FAILED
at (W %d D %d O %d OL %d) after %8.2f
seconds.\n\n", ware_num, dist_num, ord_num, oline_ol_num,
elapsed);
    fflush(stderr);
}
}

// This routine will initialize the printf format strings and replace
the
// delimiter with the one provided. The pipe symbol is the
default.
void InitFormatStrings(char delim)
{
    char *p;

    // Check if Using Default Delimiter
    if (delim == '|') return;

    // Replace Delimiters
    while (p = strchr(fmtWare, '|')) { *p = delim; }
    while (p = strchr(fmtDist, '|')) { *p = delim; }
    while (p = strchr(fmtItem, '|')) { *p = delim; }
    while (p = strchr(fmtStock, '|')) { *p = delim; }
    while (p = strchr(fmtCust, '|')) { *p = delim; }
    while (p = strchr(fmtHist, '|')) { *p = delim; }
    while (p = strchr(fmtOrder, '|')) { *p = delim; }
    while (p = strchr(fmtOLine, '|')) { *p = delim; }
    while (p = strchr(fmtNewOrd, '|')) { *p = delim; }
}

void ScalingReport(void)
{
    /* Print Scaling Values */
    fprintf(stdout, "Scaling Values in Use\n");
    fprintf(stdout, "-----\n");
    fprintf(stdout, "Warehouses: %d\n", WAREHOUSES);
    fprintf(stdout, "Districts/Warehouse: %d\n",
DISTRICTS_PER_WAREHOUSE);

```

```

    fprintf(stdout, "Customers/District: %d\n",
CUSTOMERS_PER_DISTRICT);
    fprintf(stdout, "Items: %d\n", ITEMS);
    fprintf(stdout, "Stock/Warehouse: %d\n",
STOCK_PER_WAREHOUSE);
    fprintf(stdout, "Min Order Lines/Order: %d\n",
MIN_OL_PER_ORDER);
    fprintf(stdout, "Max Order Lines/Order: %d\n",
MAX_OL_PER_ORDER);
    fprintf(stdout, "New Orders/District: %d\n",
NU_ORDERS_PER_DISTRICT);
    fprintf(stdout, "-----\n");
}

```

dbgen/tpccrnd.c

```

/*
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp.
1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
*****/

/*
 * tpccrnd.c - Random generation functions for TPC-C
 */

#include "db2tpcc.h"
#include "lval.h"

static char tbl_cust[CUSTOMERS_PER_DISTRICT];

static char alnum[] =

"0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMN
OPQRSTUVWXYZ";

static char *last_name_parts[] =
{
    "BAR",
    "OUGHT",
    "ABLE",
    "PRI",
    "PRES",
    "ESE",
    "ANTI",
    "CALLY",
    "ATION",
    "EING"
};

```

```

/*
*****
**
* seed_1_3000
*
*****
**
*/

void seed_1_3000( void )
{
    int i;

    for (i = 0; i < CUSTOMERS_PER_DISTRICT; i++) {
        tbl_cust[i] = 0;
    }
}

/*
*****
**
* random_1_3000
*
*****
**
*/

int random_1_3000( void )
{
    static int i;
    static int x;

    x = rand_integer(0, CUSTOMERS_PER_DISTRICT - 1);

    for (i = 0; i < CUSTOMERS_PER_DISTRICT; i++)
    {
        if (tbl_cust[x] == 0)
        {
            tbl_cust[x] = 1;
            return(x+1);
        } else {
            x++;
        }
        if (x == CUSTOMERS_PER_DISTRICT)
            x=0;
    }

    printf("\nfatal error in random_1_3000\n");
    abort();
}

/*
*****
**
* initialize_random
*
*****
**

```

```

*/
void initialize_random(int x, int y)
{
    srand(x);
    random(y);
}

/*
*****
**
* create_random_a_string
*
* create a random alphanumeric string, of random length
between lo and
* hi and place them in designated buffer. Routine returns the
actual
* length.
*
* parameters
* -----
* lo end of acceptable length range
* hi end of acceptable length range
*
* output
* -----
* actual length
* random alphanumeric string
*
*****
**
*/

int create_random_a_string( char *out_buffer, int length_lo, int
length_hi )
{
    int i, actual_length ;

    actual_length = rand_integer( length_lo, length_hi ) ;

    for (i = 0; i < actual_length; i++)
    {
        out_buffer[i] = alnum[rand_integer( 0, 61 )] ;
    }
    out_buffer[actual_length] = '\0' ;

    return (actual_length);
}

/*
*****
**
* create_random_n_string
*
* create a random numeric string, of random length between lo
and
* hi and place them in designated buffer. Routine returns the
actual
* length.
*

```

```

* parameters
* -----
* lo end of acceptable length range
* hi end of acceptable length range
*
* output
* -----
* actual length
* random numeric string
*
*****
**
*/

int create_random_n_string( char *out_buffer, int length_lo, int
length_hi )
{
    int i, actual_length ;

    actual_length = rand_integer( length_lo, length_hi ) ;

    for (i = 0; i < actual_length; i++)
    {
        out_buffer[i] = (char)rand_integer( 48,57 ) ;
    }
    out_buffer[actual_length] = '\0' ;

    return (actual_length);
}

/*
*****
**
* NUrand_val
*
* create a non-uniform random numeric value of type integer,
of random
* value between lo and hi. Number is NOT placed in BUFFER,
and IS
* simply RETURNED.
*
* Routine RETURNS the VALUE.
*
* parameters
* -----
* lo end of acceptable value range
* hi end of acceptable value range
*
* output
* -----
* random integer value RETURNED
*
*****
**
*/

int NUrand_val ( int A, int x, int y, int C )
{
    return((((rand_integer(0,A)|rand_integer(x,y))+C)%(y-x+1))+x);
}

```

```

/*
*****
**
* rand_integer
*
* create a uniform random numeric value of type integer, of
random
* value between lo and hi. Number is NOT placed in BUFFER,
and IS
* simply RETURNED.
*
* Routine RETURNS the VALUE.
*
* parameters
* -----
* lo end of acceptable value range
* hi end of acceptable value range
*
* output
* -----
* random integer value RETURNED
*
*****
**
*/

int rand_integer ( int val_lo, int val_hi )
{
    return((random()%(val_hi-val_lo+1))+val_lo);
}

/*
*****
**
* create_a_string_with_original
*
* create a random alphanumeric string, of random length
between lo and
* hi and place them in designated buffer. Routine returns the
actual
* length.
*
* the word "ORIGINAL" is placed at a random location in the
buffer at
* random, for a given percent of the records.
*
* percent_to_set must be an integer value from 0 to 100.
* if 0, no records will be set. If 100, all records will be set.
*
* CANNOT USE ON STRINGS OF LENGTH LESS THAN 8 !
LOWER LIMIT MUST BE > 8 !
*
* parameters
* -----
* lo end of acceptable length range
* hi end of acceptable length range
* percentage of records to set to ORIGINAL
*
* output
* -----
* actual length

```

```

* random alphanumeric string with the word "ORIGINAL" is
placed at a
* random location
*
*****
**
*/

int create_a_string_with_original( char *out_buffer, int length_lo,
                                int length_hi, int percent_to_set )
{
    int actual_length, start_pos ;

    actual_length = create_random_a_string( out_buffer,
length_lo, length_hi ) ;

    if ( rand_integer( 1, 100 ) <= percent_to_set )
    {
        start_pos = rand_integer( 0, actual_length-8 ) ;
        strncpy(out_buffer+start_pos,"ORIGINAL",8) ;
    }

    return (actual_length);
}

/*****
*
* create_random_last_name
*
* parameters:
* out_buffer - target buffer for the generated last name
*
* description:
* create_random_last_name generates a random number
from 0 to 999
* inclusive. a random name is generated by associating a
random string
* with each digit of the generated number. the three strings
are
* concatenated to generate the name
*
*****
*****/

int create_random_last_name(char *out_buffer, int cust_num)
{
    int random_num;

    if (cust_num == 0)
        random_num = NURand_val( A_C_LAST, 0, 999,
C_C_LAST_LOAD );
    else
        random_num = cust_num - 1;

    strcpy(out_buffer, last_name_parts[random_num / 100]);
    random_num %= 100;
    strcat(out_buffer, last_name_parts[random_num / 10]);
    random_num %= 10;
    strcat(out_buffer, last_name_parts[random_num]);
}

```

```

return(strlen(out_buffer));
}

include/db2tpcc.h

/*****
****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp.
1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
*****
*****/

/*
* db2tpcc.h - Macros and Miscellany
*/

#ifndef __DB2TPCC_H
#define __DB2TPCC_H

#include <sys/types.h>

#include "lval.h"

/*
*****
*/
/* Transaction Return Codes (s_transtatus)
*/
/*
*****
*/

#define INVALID_ITEM      100
#define TRAN_OK          0
#define FATAL_SQLERROR   -1

/*
*****
*/
/* Definition of Unused and Bad Items
*/
/*
*****
*/
/* Define unused item ID to be 0. This allows the SUT to
determine the
*/
/* number of items in the order as required by 2.4.1.3 and
2.4.2.2 since
*/
/* the assumption that any item with OL_I_ID = 0 is unused will
be true.
*/

```

```

/* This in turn requires that the value used for an invalid item is
*/
/* equal to ITEMS + 1.
*/
/*
*****
*/

#define INVALID_ITEM_ID (2 * ITEMS) + 1
#define UNUSED_ITEM_ID 0

#define MIN_WAREHOUSE 1
#define MAX_WAREHOUSE WAREHOUSES

/*****
*****/
/* NURand Constants
*/
/* C_C_LAST_RUN and C_C_LAST_LOAD must adhere to
clause 2.1.6.
*/
/* Analysis indicates that a C_LAST delta of 85 is optimal.
*/
/*****
*****/
#define C_C_LAST_RUN      88
#define C_C_LAST_LOAD    173
#define C_C_ID           319
#define C_OL_I_ID        3849
#define A_C_LAST         255
#define A_C_ID           1023
#define A_OL_I_ID        8191

/*****
*****/
/* Transaction Type Identifiers
*/
/*****
*****/

#define CLIENT_SQL      0
#define NEWORD_SQL     1
#define PAYMENT_SQL     2
#define ORDSTAT_SQL    3
#define DELIVERY_SQL   4
#define STOCKLEV_SQL   5

#define SPGENERAL_PAD 3
#define SPGENERAL_ADJUST sizeof(int16_t)

struct in_neword_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    struct in_items_struct {
        int32_t s_OL_I_ID;
        int32_t s_OL_SUPPLY_W_ID;
        int16_t s_OL_QUANTITY;
        int16_t pad1[3];
    } in_item[15];
    int64_t s_O_ENTRY_D_time; /* init by SUT */
    int32_t s_C_ID;
    int32_t s_W_ID;
    int16_t s_D_ID;
    int16_t s_O_OL_CNT; /* init by SUT */
    int16_t s_all_local;
    int16_t duplicate_items;
}

```

```

);

struct out_neword_struct {
  int16_t len;
  int16_t pad[SPGENERAL_PAD];
  struct items_struct {
    int32_t s_I_PRICE;
    int32_t s_OL_AMOUNT;
    int16_t s_S_QUANTITY;
    int16_t pad2;
    char s_I_NAME[25];
    char s_brand_generic;
  } item[15];
  int64_t s_O_ENTRY_D_time;
  int32_t s_W_TAX;
  int32_t s_D_TAX;
  int32_t s_C_DISCOUNT;
  int32_t s_total_amount;
  int32_t s_O_ID;
  int16_t s_O_OL_CNT;
  int16_t s_transtatus;
  int16_t deadlocks;
  char s_C_LAST[17];
  char s_C_CREDIT[3];
};

struct in_payment_struct {
  int16_t len;
  int16_t pad[SPGENERAL_PAD];
  int64_t s_H_DATE_time; /* init by SUT */
  int64_t s_H_AMOUNT;
  int32_t s_W_ID;
  int32_t s_C_W_ID;
  int32_t s_C_ID;
  int16_t s_C_D_ID;
  int16_t s_D_ID;
  char s_C_LAST[17];
};

struct out_payment_struct {
  int16_t len;
  int16_t pad[SPGENERAL_PAD];
  int64_t s_H_DATE_time;
  int64_t s_C_SINCE_time;
  int64_t s_C_CREDIT_LIM;
  int64_t s_C_BALANCE;
  int32_t s_C_DISCOUNT;
  int32_t s_C_ID;
  int16_t s_transtatus;
  int16_t deadlocks;
  char s_W_STREET_1[21];
  char s_W_STREET_2[21];
  char s_W_CITY[21];
  char s_W_STATE[3];
  char s_W_ZIP[10];
  char s_D_STREET_1[21];
  char s_D_STREET_2[21];
  char s_D_CITY[21];
  char s_D_STATE[3];
  char s_D_ZIP[10];
  char s_C_FIRST[17];
  char s_C_MIDDLE[3];
};

```

```

char s_C_LAST[17];
char s_C_STREET_1[21];
char s_C_STREET_2[21];
char s_C_CITY[21];
char s_C_STATE[3];
char s_C_ZIP[10];
char s_C_PHONE[17];
char s_C_CREDIT[3];
char s_C_DATA[201];
};

struct in_ordstat_struct {
  int16_t len;
  int16_t pad[SPGENERAL_PAD];
  int32_t s_C_ID;
  int32_t s_W_ID;
  int16_t s_D_ID;
  int16_t pad1[3];
  char s_C_LAST[17];
};

struct out_ordstat_struct {
  int16_t len;
  int16_t pad[SPGENERAL_PAD];
  int64_t s_C_BALANCE;
  int64_t s_O_ENTRY_D_time;
  int32_t s_C_ID;
  int32_t s_O_ID;
  int16_t s_O_CARRIER_ID;
  int16_t s_ol_cnt;
  int16_t pad1[2];
  struct oitems_struct {
    int64_t s_OL_DELIVERY_D_time;
    int32_t s_OL_AMOUNT;
    int32_t s_OL_I_ID;
    int32_t s_OL_SUPPLY_W_ID;
    int16_t s_OL_QUANTITY;
    int16_t pad2;
  } item[15];
  int16_t s_transtatus;
  int16_t deadlocks;
  char s_C_FIRST[17];
  char s_C_MIDDLE[3];
  char s_C_LAST[17];
};

struct in_delivery_struct {
  int16_t len;
  int16_t pad[SPGENERAL_PAD];
  int64_t s_O_DELIVERY_D_time; /* init by SUT */
  int32_t s_W_ID;
  int16_t s_O_CARRIER_ID;
};

struct out_delivery_struct {
  int16_t len;
  int16_t pad[SPGENERAL_PAD];
  int32_t s_O_ID[10];
  int16_t s_transtatus;
  int16_t deadlocks;
};

```

```

struct in_stocklev_struct {
  int16_t len;
  int16_t pad[SPGENERAL_PAD];
  int32_t s_threshold;
  int32_t s_W_ID;
  int16_t s_D_ID;
};

struct out_stocklev_struct {
  int16_t len;
  int16_t pad[SPGENERAL_PAD];
  int32_t s_low_stock;
  int16_t s_transtatus;
  int16_t deadlocks;
};

/*
*****
*/
/* Transaction Prototypes */
/*
*****
*/

#ifdef __cplusplus
extern "C" {
#endif

extern int neword_sql(struct in_neword_struct*, struct
out_neword_struct*);
extern int payment_sql(struct in_payment_struct*, struct
out_payment_struct*);
extern int ordstat_sql(struct in_ordstat_struct*, struct
out_ordstat_struct*);
extern int delivery_sql(struct in_delivery_struct*, struct
out_delivery_struct*);
extern int stocklev_sql(struct in_stocklev_struct*, struct
out_stocklev_struct*);

#ifdef __cplusplus
}
#endif

/*
*****
*/
/* DB2 Connect/Disconnect & Thread Context Wrappers */
/*
*****
*/

#ifdef __cplusplus
extern "C" {
#endif

extern int connect_to_TM(char*);
extern int connect_to_TM_auth(char*, char*, char*);
extern int disconnect_from_TM(void);

#ifdef __cplusplus

```

```

}
#endif

#endif // __DB2TPCC_H

```

include/lval.h

```

#ifndef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 81600
#define DISTRICTS_PER_WAREHOUSE 10
#define CUSTOMERS_PER_DISTRICT 3000
#define ITEMS 100000
#define STOCK_PER_WAREHOUSE 100000
#define MIN_OL_PER_ORDER 5
#define MAX_OL_PER_ORDER 15
#define NU_ORDERS_PER_DISTRICT 900
#endif // __LVAL_H

```

include/platform.h

```

/*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp.
** 1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
** IBM Corp.
*****/

/*
 * platform.h - Platform Isolation Layer
 */

#ifndef __PLATFORM_H
#define __PLATFORM_H

/*
*****
*/
/* Generic Macros */
/*
*****
*/
#define GEN_ERRCODE errno

/*
*****
*/
/* Windows I/O Macros */
/*
*****
*/
/*
*****
*/
/* UNIX I/O Macros */

```

```

/*
*****
*/
#include <fcntl.h>

#define IOH_INIT(hnd, type, name) \
    hnd->fd = -1; \
    hnd->type = type; \
    hnd->name = name;

#define IOH_CREATE(hnd) \
    if (hnd->type == IOH_PIPE) { \
        rc = mkfifo(hnd->name, 0666); \
    } else { \
        rc = 0; \
    }

#define IOH_OPEN(hnd) \
    if (hnd->type == IOH_FILE_APPEND) \
    { \
        hnd->fd = open(hnd->name, O_WRONLY | O_CREAT | \
O_APPEND, 0666); \
    } else { \
        hnd->fd = open(hnd->name, O_WRONLY | O_CREAT | \
O_TRUNC, 0666); \
    } \
    if (hnd->fd == -1) { \
        rc = -1; \
    } else { \
        rc = 0; \
    }

#define IOH_WRITE(hnd, buff, num, num2) \
    rc = write(hnd->fd, buff, num); \
    if (rc >= 0) { \
        num2 = rc; \
        rc = 0; \
    }

#define IOH_FLUSH(hnd) rc = 0;
#define IOH_CLOSE(hnd) rc = close(hnd->fd);
#define IOH_DELETE(hnd) if (hnd->type == IOH_PIPE) { rc = \
unlink(hnd->name); }

typedef unsigned int IOH_NUM;
typedef int IOH_HND;

/*
*****
**/
/* UNIX Semaphore Macros */
/*
*****
**/
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>

union semun {
    int val;
    struct semid_ds *buf;
    unsigned short int *array;
} semUnion;

```

```

struct sembuf semBuf;

#define SEM_HANDLE int

#define SEM_INIT(hnd, x, name) \
    if ( (hnd = semget(IPC_PRIVATE, 1, IPC_CREAT | \
IPC_EXCL | S_IRUSR | S_IWUSR | S_IRGRP | S_IWGRP | \
S_IROTH | S_IWOTH)) == -1) \
        API_ERROR(__LINE__, "semget", (rc=GEN_ERRCODE)); \
    semUnion.val = x; \
    if ( semctl(hnd, 0, SETVAL, semUnion) < 0 ) \
        API_ERROR(__LINE__, "semctl SETVAL", \
(rc=GEN_ERRCODE));

#define SEM_WAIT(hnd) \
    semBuf.sem_num = 0; \
    semBuf.sem_op = -1; \
    semBuf.sem_flg = SEM_UNDO; \
    if ( semop(hnd, &semBuf, 1) < 0 ) \
        API_ERROR(__LINE__, "semop wait", \
(rc=GEN_ERRCODE));

#define SEM_FREE(hnd) \
    semBuf.sem_num = 0; \
    semBuf.sem_op = 1; \
    semBuf.sem_flg = SEM_UNDO; \
    if ( semop(hnd, &semBuf, 1) < 0 ) \
        API_ERROR(__LINE__, "semop free", \
(rc=GEN_ERRCODE));

#define SEM_DESTROY(hnd) \
    if ( semctl(hnd, 0, IPC_RMID, 0) ) \
        API_ERROR(__LINE__, "semctl IPC_RMID", \
(rc=GEN_ERRCODE));

/*
*****
*/
/* Common I/O Macros and Definitions */
/*
*****
*/
#define IOH_FILE 1
#define IOH_PIPE 2
#define IOH_FILE_APPEND 3

#define IOH_ERRMSG(hnd, msg) \
    if (rc != 0) { \
        fprintf(stderr, "Error %d %s fd %d (%d, %s)\n", \
GEN_ERRCODE, msg, \
        hnd->fd, hnd->type, hnd->name); \
        return rc; \
    }

struct _ioh {
    IOH_HND fd;
    int type;
    char *name;
};

typedef struct _ioh ioHandle;

```

```

/*
*****/
/* Generic I/O Routine Prototypes */
/*
*****/
int GenericOpen(ioHandle *hnd, int type, char *name);
int GenericWrite(ioHandle *hnd, char *Buffer, unsigned int
numBytes);
int GenericClose(ioHandle *hnd);

/*
*****/
/* Generic I/O Routines */
/*
*****/
int GenericOpen(ioHandle *hnd, int type, char *name)
{
    int rc = 0;

    IOH_INIT(hnd, type, name)

    IOH_CREATE(hnd)
    IOH_ERRMSG(hnd, "creating")

    IOH_OPEN(hnd)
    IOH_ERRMSG(hnd, "opening")

    return rc;
}

int GenericWrite(ioHandle *hnd, char *Buffer, unsigned int
numBytes)
{
    int rc = 0;
    int numBytesWritten = -1;

    IOH_WRITE(hnd, Buffer, numBytes, numBytesWritten)
    IOH_ERRMSG(hnd, "writing")
    if (numBytes != numBytesWritten) {
        fprintf(stderr, "Truncated data writing to fd %d (%d, %s)\n",
hnd->fd, hnd->type, hnd->name);
        rc = -1;
    }
    return rc;
}

int GenericClose(ioHandle *hnd)
{
    int rc = 0;

    IOH_FLUSH(hnd)
    IOH_ERRMSG(hnd, "flushing")

    IOH_CLOSE(hnd)
    IOH_ERRMSG(hnd, "closing")

    IOH_DELETE(hnd)
    IOH_ERRMSG(hnd, "deleting")

    return rc;
}

```

```

#endif // __PLATFORM_H

include/tpccrnd.h

/*
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp.
1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
*****/

/*
* tpccrnd.h - Random generation functions for TPC-C
*/

#ifndef __TPCCRND_H
#define __TPCCRND_H

void initialize_random(int x, int y);
int rand_integer( int val_lo, int val_hi ) ;
int NUrnd_val( int A, int val_lo, int val_hi, int C ) ;

void seed_1_3000( void );
int random_1_3000( void );

int create_random_a_string( char *out_buffer,
int length_lo,
int length_hi );
int create_random_n_string( char *out_buffer,
int length_lo,
int length_hi );
int create_a_string_with_original( char *out_buffer,
int length_lo,
int length_hi,
int percent_to_set );
int create_random_last_name(char *out_buffer, int cust_num);

#endif // __TPCCRND_H

tpccenv.sh

#####
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp.
1996 - 2006
## All Rights Reserved.

```

```

##
## US Government Users Restricted Rights - Use, duplication
or
## disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
#####
#####

#
# tpccenv.sh - UNIX Environment Setup
#

# The Kit Version
export TPCC_VERSION=CK040318

# The DB2 Instance Name (for DB2)
export DB2INSTANCE=${USER}

# The OS being used (i.e. "UNIX", "WINDOWS")
export PLATFORM=UNIX

# The type of make command and slash used by the OS.
# (i.e. UNIX - "/", WINDOWS - "\").
# These are referenced all over the kit.
export SLASH="/";
export MAKE=make

# Specifies whether or not to use dari stored proc's for the TPC-
C driver. Set to either DARIVERSION or NONDARI;
#export TPCC_SPTYPE=NOSP
#export TPCC_SPTYPE=SPGENERAL2
export TPCC_SPTYPE=SPGENERAL
#export TPCC_SPTYPE=DARI2SQLDA

export DB2VERSION=v8

# The schema name is typically the SQL authorization ID (or
username).
# This is required for runstats and EEE.
export TPCC_SCHEMA=${USER}

# DB2 EE/EEE Configuration
export DB2EDITION=EE
#export DB2EDITION=EEE
export DB2NODE=0
export DB2NODES=1; # set to the number of nodes you
have. Set to 1 for EE.

# TPCC General Configuration
export TPCC_DBNAME=TPCC
export TPCC_ROOT=${HOME}/tpc-c.ibm
export TPCC_SQLLIB=${HOME}/sqllib
export TPCC_RUNDATA=${HOME}/tpccdata

# TPCC Debug Configuration
# This is the path where all error and debug logs are placed.
# To get debugging from within the stored procedures, you must
# set DB2ENVLIST="TPCC_DEBUGDIR" in tpcc.config.
export TPCC_DEBUGDIR=/tmp

# Specifies where stored procedures should be placed and if
they should

```

```
# be fenced.  
export TPCC_SPDIR=${TPCC_SQLLIB}/function  
export TPCC_FENCED=NO
```

Appendix - D: RTE Scripts

D.1 RTE Parameters

rteparams

```
#define MASTER_NUM1 0
#define MASTER_NUM2 0
#define MASTER_NUM3 0
#define MASTER_NUM4 0
#define MASTER_NUM5 1
#define MASTER_NUM6 0
#define MASTER_NUM7 0
#define MASTER_NUM8 0
#define MASTER_NUM9 0
#define MASTER_NUM10 0

#if MASTER_NUM1
MASTER "master1"
#elif MASTER_NUM2
MASTER "master2"
#elif MASTER_NUM3
MASTER "master3"
#elif MASTER_NUM4
MASTER "master4"
#elif MASTER_NUM5
MASTER "master5"
#elif MASTER_NUM6
MASTER "master6"
#elif MASTER_NUM7
MASTER "master7"
#elif MASTER_NUM8
MASTER "master8"
#elif MASTER_NUM9
MASTER "master9"
#elif MASTER_NUM10
MASTER "master10"
#endif
/*---- SUT -----*/
SUT="toraus"
/*-----*/
LASTC=86
MEASUREMENT="1"
WAREHOUSES=81600
/*----- SLAVES -----*/

#if MASTER_NUM1
SLAVES driver1a, driver1b, driver1c, driver1d, driver1e,
driver1f, driver1g, driver1h, driver2a, driver2b, driver2c,
driver2d, driver2e, driver2f, driver2g, driver2h, driver3a,
driver3b, driver3c, driver3d, driver3e, driver3f, driver3g,
driver3h, driver4a, driver4b, driver4c, driver4d, driver4e,
driver4f, driver4g, driver4h
#elif MASTER_NUM2
SLAVES driver5a, driver5b, driver5c, driver5d, driver5e,
driver5f, driver5g, driver5h, driver6a, driver6b, driver6c,
driver6d, driver6e, driver6f, driver6g, driver6h, driver7a,
driver7b, driver7c, driver7d, driver7e, driver7f, driver7g,
```

```
driver7h, driver8a, driver8b, driver8c, driver8d, driver8e,
driver8f, driver8g, driver8h
#elif MASTER_NUM3
SLAVES driver9a, driver9b, driver9c, driver9d, driver9e,
driver9f, driver9g, driver9h, driver10a, driver10b, driver10c,
driver10d, driver10e, driver10f, driver10g, driver10h, driver11a,
driver11b, driver11c, driver11d, driver11e, driver11f, driver11g,
driver11h, driver12a, driver12b, driver12c, driver12d, driver12e,
driver12f, driver12g, driver12h
#elif MASTER_NUM4
SLAVES driver13a, driver13b, driver13c, driver13d, driver13e,
driver13f, driver13g, driver13h, driver14a, driver14b, driver14c,
driver14d, driver14e, driver14f, driver14g, driver14h, driver15a,
driver15b, driver15c, driver15d, driver15e, driver15f, driver15g,
driver15h, driver16a, driver16b, driver16c, driver16d, driver16e,
driver16f, driver16g, driver16h
#elif MASTER_NUM5
SLAVES driver17a, driver17b, driver17c, driver17d, driver17e,
driver17f, driver17g, driver17h, driver18a, driver18b, driver18c,
driver18d, driver18e, driver18f, driver18g, driver18h, driver19a,
driver19b, driver19c, driver19d, driver19e, driver19f, driver19g,
driver19h, driver20a, driver20b, driver20c, driver20d, driver20e,
driver20f, driver20g, driver20h
#elif MASTER_NUM6
SLAVES driver21a, driver21b, driver21c, driver21d, driver21e,
driver21f, driver21g, driver21h, driver22a, driver22b, driver22c,
driver22d, driver22e, driver22f, driver22g, driver22h, driver23a,
driver23b, driver23c, driver23d, driver23e, driver23f, driver23g,
driver23h, driver24a, driver24b, driver24c, driver24d, driver24e,
driver24f, driver24g, driver24h
#elif MASTER_NUM7
SLAVES driver25a, driver25b, driver25c, driver25d, driver25e,
driver25f, driver25g, driver25h, driver26a, driver26b, driver26c,
driver26d, driver26e, driver26f, driver26g, driver26h, driver27a,
driver27b, driver27c, driver27d, driver27e, driver27f, driver27g,
driver27h, driver28a, driver28b, driver28c, driver28d, driver28e,
driver28f, driver28g, driver28h
#elif MASTER_NUM8
SLAVES driver29a, driver29b, driver29c, driver29d, driver29e,
driver29f, driver29g, driver29h, driver30a, driver30b, driver30c,
driver30d, driver30e, driver30f, driver30g, driver30h, driver31a,
driver31b, driver31c, driver31d, driver31e, driver31f, driver31g,
driver31h, driver32a, driver32b, driver32c, driver32d, driver32e,
driver32f, driver32g, driver32h
#elif MASTER_NUM9
SLAVES driver33a, driver33b, driver33c, driver33d, driver33e,
driver33f, driver33g, driver33h, driver34a, driver34b, driver34c,
driver34d, driver34e, driver34f, driver34g, driver34h, driver35a,
driver35b, driver35c, driver35d, driver35e, driver35f, driver35g,
driver35h, driver36a, driver36b, driver36c, driver36d, driver36e,
driver36f, driver36g, driver36h
#elif MASTER_NUM10
SLAVES driver37a, driver37b, driver37c, driver37d, driver37e,
driver37f, driver37g, driver37h, driver38a, driver38b, driver38c,
driver38d, driver38e, driver38f, driver38g, driver38h, driver39a,
driver39b, driver39c, driver39d, driver39e, driver39f, driver39g,
driver39h, driver40a, driver40b, driver40c, driver40d, driver40e,
driver40f, driver40g, driver40h
#endif

/* ---- CLIENTS -----*/
#if MASTER_NUM1
```

```
MAIN_CLIENT = client1
CLIENT_REAL = "client1 client2 client3 client4"
#elif MASTER_NUM2
MAIN_CLIENT = client5
CLIENT_REAL = "client5 client6 client7 client8"
#elif MASTER_NUM3
MAIN_CLIENT = client9
CLIENT_REAL = "client9 client10 client11 client12"
#elif MASTER_NUM4
MAIN_CLIENT = client13
CLIENT_REAL = "client13 client14 client15 client16"
#elif MASTER_NUM5
MAIN_CLIENT = client17
CLIENT_REAL = "client17 client18 client19 client20"
#elif MASTER_NUM6
MAIN_CLIENT = client21
CLIENT_REAL = "client21 client22 client23 client24"
#elif MASTER_NUM7
MAIN_CLIENT = client25
CLIENT_REAL = "client25 client26 client27 client28"
#elif MASTER_NUM8
MAIN_CLIENT = client29
CLIENT_REAL = "client29 client30 client31 client32"
#elif MASTER_NUM9
MAIN_CLIENT = client33
CLIENT_REAL = "client33 client34 client35 client36"
#elif MASTER_NUM10
MAIN_CLIENT = client37
CLIENT_REAL = "client37 client38 client39 client40"
#endif

/*---- more client ctuff -----*/
#if MASTER_NUM1
CLIENT client1a tpcc tpcc
CLIENT client1b tpcc tpcc

CLIENT client2a tpcc tpcc
CLIENT client2b tpcc tpcc

CLIENT client3a tpcc tpcc
CLIENT client3b tpcc tpcc

CLIENT client4a tpcc tpcc
CLIENT client4b tpcc tpcc

#elif MASTER_NUM2
CLIENT client5a tpcc tpcc
CLIENT client5b tpcc tpcc

CLIENT client6a tpcc tpcc
CLIENT client6b tpcc tpcc

CLIENT client7a tpcc tpcc
CLIENT client7b tpcc tpcc

CLIENT client8a tpcc tpcc
CLIENT client8b tpcc tpcc

#elif MASTER_NUM3
CLIENT client9a tpcc tpcc
CLIENT client9b tpcc tpcc
```



```

CLIENT client10a tpcc tpcc
CLIENT client10b tpcc tpcc

CLIENT client11a tpcc tpcc
CLIENT client11b tpcc tpcc

CLIENT client12a tpcc tpcc
CLIENT client12b tpcc tpcc

#elif MASTER_NUM4
CLIENT client13a tpcc tpcc
CLIENT client13b tpcc tpcc

CLIENT client14a tpcc tpcc
CLIENT client14b tpcc tpcc

CLIENT client15a tpcc tpcc
CLIENT client15b tpcc tpcc

CLIENT client16a tpcc tpcc
CLIENT client16b tpcc tpcc

#elif MASTER_NUM5
CLIENT client17a tpcc tpcc
CLIENT client17b tpcc tpcc

CLIENT client18a tpcc tpcc
CLIENT client18b tpcc tpcc

CLIENT client19a tpcc tpcc
CLIENT client19b tpcc tpcc

CLIENT client20a tpcc tpcc
CLIENT client20b tpcc tpcc

#elif MASTER_NUM6
CLIENT client21a tpcc tpcc
CLIENT client21b tpcc tpcc

CLIENT client22a tpcc tpcc
CLIENT client22b tpcc tpcc

CLIENT client23a tpcc tpcc
CLIENT client23b tpcc tpcc

CLIENT client24a tpcc tpcc
CLIENT client24b tpcc tpcc

#elif MASTER_NUM7
CLIENT client25a tpcc tpcc
CLIENT client25b tpcc tpcc

CLIENT client26a tpcc tpcc
CLIENT client26b tpcc tpcc

CLIENT client27a tpcc tpcc
CLIENT client27b tpcc tpcc

CLIENT client28a tpcc tpcc
CLIENT client28b tpcc tpcc

#elif MASTER_NUM8

```

```

CLIENT client29a tpcc tpcc
CLIENT client29b tpcc tpcc

CLIENT client30a tpcc tpcc
CLIENT client30b tpcc tpcc

CLIENT client31a tpcc tpcc
CLIENT client31b tpcc tpcc

CLIENT client32a tpcc tpcc
CLIENT client32b tpcc tpcc

#elif MASTER_NUM9
CLIENT client33a tpcc tpcc
CLIENT client33b tpcc tpcc

CLIENT client34a tpcc tpcc
CLIENT client34b tpcc tpcc

CLIENT client35a tpcc tpcc
CLIENT client35b tpcc tpcc

CLIENT client36a tpcc tpcc
CLIENT client36b tpcc tpcc

#elif MASTER_NUM10
CLIENT client37a tpcc tpcc
CLIENT client37b tpcc tpcc

CLIENT client38a tpcc tpcc
CLIENT client38b tpcc tpcc

CLIENT client39a tpcc tpcc
CLIENT client39b tpcc tpcc

CLIENT client40a tpcc tpcc
CLIENT client40b tpcc tpcc

#endif
/*-----*/
TELNET telnet 23
SOCKET socket 199703
/* --- Sockets -----*/
#if MASTER_NUM1
SOCKET_NETWORK socket1 80 driver1a
SOCKET_NETWORK socket2 80 driver1b
SOCKET_NETWORK socket3 80 driver1c
SOCKET_NETWORK socket4 80 driver1d
SOCKET_NETWORK socket5 80 driver1e
SOCKET_NETWORK socket6 80 driver1f
SOCKET_NETWORK socket7 80 driver1g
SOCKET_NETWORK socket8 80 driver1h
SOCKET_NETWORK socket9 80 driver2a
SOCKET_NETWORK socket10 80 driver2b
SOCKET_NETWORK socket11 80 driver2c
SOCKET_NETWORK socket12 80 driver2d
SOCKET_NETWORK socket13 80 driver2e
SOCKET_NETWORK socket14 80 driver2f
SOCKET_NETWORK socket15 80 driver2g
SOCKET_NETWORK socket16 80 driver2h
SOCKET_NETWORK socket17 80 driver3a
SOCKET_NETWORK socket18 80 driver3b

```

```

SOCKET_NETWORK socket19 80 driver3c
SOCKET_NETWORK socket20 80 driver3d
SOCKET_NETWORK socket21 80 driver3e
SOCKET_NETWORK socket22 80 driver3f
SOCKET_NETWORK socket23 80 driver3g
SOCKET_NETWORK socket24 80 driver3h
SOCKET_NETWORK socket25 80 driver4a
SOCKET_NETWORK socket26 80 driver4b
SOCKET_NETWORK socket27 80 driver4c
SOCKET_NETWORK socket28 80 driver4d
SOCKET_NETWORK socket29 80 driver4e
SOCKET_NETWORK socket30 80 driver4f
SOCKET_NETWORK socket31 80 driver4g
SOCKET_NETWORK socket32 80 driver4h
#elif MASTER_NUM2
SOCKET_NETWORK socket33 80 driver5a
SOCKET_NETWORK socket34 80 driver5b
SOCKET_NETWORK socket35 80 driver5c
SOCKET_NETWORK socket36 80 driver5d
SOCKET_NETWORK socket37 80 driver5e
SOCKET_NETWORK socket38 80 driver5f
SOCKET_NETWORK socket39 80 driver5g
SOCKET_NETWORK socket40 80 driver5h
SOCKET_NETWORK socket41 80 driver6a
SOCKET_NETWORK socket42 80 driver6b
SOCKET_NETWORK socket43 80 driver6c
SOCKET_NETWORK socket44 80 driver6d
SOCKET_NETWORK socket45 80 driver6e
SOCKET_NETWORK socket46 80 driver6f
SOCKET_NETWORK socket47 80 driver6g
SOCKET_NETWORK socket48 80 driver6h
SOCKET_NETWORK socket49 80 driver7a
SOCKET_NETWORK socket50 80 driver7b
SOCKET_NETWORK socket51 80 driver7c
SOCKET_NETWORK socket52 80 driver7d
SOCKET_NETWORK socket53 80 driver7e
SOCKET_NETWORK socket54 80 driver7f
SOCKET_NETWORK socket55 80 driver7g
SOCKET_NETWORK socket56 80 driver7h
SOCKET_NETWORK socket57 80 driver8a
SOCKET_NETWORK socket58 80 driver8b
SOCKET_NETWORK socket59 80 driver8c
SOCKET_NETWORK socket60 80 driver8d
SOCKET_NETWORK socket61 80 driver8e
SOCKET_NETWORK socket62 80 driver8f
SOCKET_NETWORK socket63 80 driver8g
SOCKET_NETWORK socket64 80 driver8h
#elif MASTER_NUM3
SOCKET_NETWORK socket65 80 driver9a
SOCKET_NETWORK socket66 80 driver9b
SOCKET_NETWORK socket67 80 driver9c
SOCKET_NETWORK socket68 80 driver9d
SOCKET_NETWORK socket69 80 driver9e
SOCKET_NETWORK socket70 80 driver9f
SOCKET_NETWORK socket71 80 driver9g
SOCKET_NETWORK socket72 80 driver9h
SOCKET_NETWORK socket73 80 driver10a
SOCKET_NETWORK socket74 80 driver10b
SOCKET_NETWORK socket75 80 driver10c
SOCKET_NETWORK socket76 80 driver10d
SOCKET_NETWORK socket77 80 driver10e
SOCKET_NETWORK socket78 80 driver10f

```



```

SOCKET_NETWORK socket259 80 driver33c
SOCKET_NETWORK socket260 80 driver33d
SOCKET_NETWORK socket261 80 driver33e
SOCKET_NETWORK socket262 80 driver33f
SOCKET_NETWORK socket263 80 driver33g
SOCKET_NETWORK socket264 80 driver33h
SOCKET_NETWORK socket265 80 driver34a
SOCKET_NETWORK socket266 80 driver34b
SOCKET_NETWORK socket267 80 driver34c
SOCKET_NETWORK socket268 80 driver34d
SOCKET_NETWORK socket269 80 driver34e
SOCKET_NETWORK socket270 80 driver34f
SOCKET_NETWORK socket271 80 driver34g
SOCKET_NETWORK socket272 80 driver34h
SOCKET_NETWORK socket273 80 driver35a
SOCKET_NETWORK socket274 80 driver35b
SOCKET_NETWORK socket275 80 driver35c
SOCKET_NETWORK socket276 80 driver35d
SOCKET_NETWORK socket277 80 driver35e
SOCKET_NETWORK socket278 80 driver35f
SOCKET_NETWORK socket279 80 driver35g
SOCKET_NETWORK socket280 80 driver35h
SOCKET_NETWORK socket281 80 driver36a
SOCKET_NETWORK socket282 80 driver36b
SOCKET_NETWORK socket283 80 driver36c
SOCKET_NETWORK socket284 80 driver36d
SOCKET_NETWORK socket285 80 driver36e
SOCKET_NETWORK socket286 80 driver36f
SOCKET_NETWORK socket287 80 driver36g
SOCKET_NETWORK socket288 80 driver36h
#elif MASTER_NUM10
SOCKET_NETWORK socket289 80 driver37a
SOCKET_NETWORK socket290 80 driver37b
SOCKET_NETWORK socket291 80 driver37c
SOCKET_NETWORK socket292 80 driver37d
SOCKET_NETWORK socket293 80 driver37e
SOCKET_NETWORK socket294 80 driver37f
SOCKET_NETWORK socket295 80 driver37g
SOCKET_NETWORK socket296 80 driver37h
SOCKET_NETWORK socket297 80 driver38a
SOCKET_NETWORK socket298 80 driver38b
SOCKET_NETWORK socket299 80 driver38c
SOCKET_NETWORK socket300 80 driver38d
SOCKET_NETWORK socket301 80 driver38e
SOCKET_NETWORK socket302 80 driver38f
SOCKET_NETWORK socket303 80 driver38g
SOCKET_NETWORK socket304 80 driver38h
SOCKET_NETWORK socket305 80 driver39a
SOCKET_NETWORK socket306 80 driver39b
SOCKET_NETWORK socket307 80 driver39c
SOCKET_NETWORK socket308 80 driver39d
SOCKET_NETWORK socket309 80 driver39e
SOCKET_NETWORK socket310 80 driver39f
SOCKET_NETWORK socket311 80 driver39g
SOCKET_NETWORK socket312 80 driver39h
SOCKET_NETWORK socket313 80 driver40a
SOCKET_NETWORK socket314 80 driver40b
SOCKET_NETWORK socket315 80 driver40c
SOCKET_NETWORK socket316 80 driver40d
SOCKET_NETWORK socket317 80 driver40e
SOCKET_NETWORK socket318 80 driver40f
SOCKET_NETWORK socket319 80 driver40g

```

```

SOCKET_NETWORK socket320 80 driver40h
#endif
/*-----*/
OUTPUTNAME="regattaH"
CPU=48
#if 0
BEGIN_WAIT=5:00
RAMPUP=42:30
RUNTIME=30:00
RAMPDOWN_WAIT=5:00
RAMPDOWN=17:00
#else
BEGIN_WAIT=25:00
RAMPUP=30:00
RUNTIME=150:00
RAMPDOWN_WAIT=2:00
RAMPDOWN=20:00
#endif
/*RAMPUP_SEC 1800 */
/*RUNTIME_SEC 9000 */
/*WAREHOUSES 81600 */
/*CLIENTS 1480 */
INTERVAL=1:00 /* Interval to calculate mix from */
LOGIN_MAX_LOAD = 8
LOGIN_BEGIN = 0 /* skip login state if set to 1 */
NOBEGIN = 1
KEYSTROKE_PACKET_SIZE = 0
MAX_CONCURRENT_SPAWN = 5
SPAWN_COUNT = 4
MIN_PORT = 8088
MAX_PORT = 8089
/* User variables. Think, Emulex Delay, %desired, %min, %max */
#if 1 /* Testing */
NEWORDER = "12.02, 0, 0"
PAYMENT = "12.02, 0, 0, 43.03, 43.03, 43.03 "
ORDSTAT = "10.01, 0, 0, 4.02, 4.02, 4.02 "
DELIVERY = "05.02, 0, 0, 4.02, 4.02, 4.02 "
STOCKLEV = "05.02, 0, 0, 4.02, 4.02, 4.02 "
#elseif 0 /* From rtparams.null */
NEWORDER = "12.25, 0.42, 0.38"
PAYMENT = "12.25, 0.19, 0.23, 43.2, 41.1, 45.3 "
ORDSTAT = "10.50, 0.39, 0.21, 4.1, 3.9, 4.3 "
DELIVERY = "05.5, 0.19, 0.15, 4.1, 3.9, 4.3 "
STOCKLEV = "05.5, 0.25, 0.18, 4.1, 3.9, 4.3 "
#elseif 0 /* From Pookeepsie */
NEWORDER = "16.25, 0.42, 0.38"
PAYMENT = "16.25, 0.19, 0.23, 43.15, 43.15, 43.15 "
ORDSTAT = "14.50, 0.39, 0.21, 4.03, 4.03, 4.03 "
DELIVERY = "09.50, 0.19, 0.15, 4.03, 4.03, 4.03 "
STOCKLEV = "09.50, 0.25, 0.18, 4.03, 4.03, 4.03 "
#endif
/*---- Starting users on sockets -----*/
#if MASTER_NUM1
START_RANGE client1a socket1 2550 0-255
START_RANGE client1a socket2 2550 255-510
START_RANGE client1a socket3 2550 510-765
START_RANGE client1a socket4 2550 765-1020

START_RANGE client1a socket5 2550 1020-1275
START_RANGE client1a socket6 2550 1275-1530
START_RANGE client1a socket7 2550 1530-1785

```

```

START_RANGE client1a socket8 2550 1785-2040

START_RANGE client2a socket9 2550 2040-2295
START_RANGE client2a socket10 2550 2295-2550
START_RANGE client2a socket11 2550 2550-2805
START_RANGE client2a socket12 2550 2805-3060

START_RANGE client2a socket13 2550 3060-3315
START_RANGE client2a socket14 2550 3315-3570
START_RANGE client2a socket15 2550 3570-3825
START_RANGE client2a socket16 2550 3825-4080

START_RANGE client3a socket17 2550 4080-4335
START_RANGE client3a socket18 2550 4335-4590
START_RANGE client3a socket19 2550 4590-4845
START_RANGE client3a socket20 2550 4845-5100

START_RANGE client3a socket21 2550 5100-5355
START_RANGE client3a socket22 2550 5355-5610
START_RANGE client3a socket23 2550 5610-5865
START_RANGE client3a socket24 2550 5865-6120

START_RANGE client4a socket25 2550 6120-6375
START_RANGE client4a socket26 2550 6375-6630
START_RANGE client4a socket27 2550 6630-6885
START_RANGE client4a socket28 2550 6885-7140

START_RANGE client4a socket29 2550 7140-7395
START_RANGE client4a socket30 2550 7395-7650
START_RANGE client4a socket31 2550 7650-7905
START_RANGE client4a socket32 2550 7905-8160

#elif MASTER_NUM2
START_RANGE client5a socket33 2550 8160-8415
START_RANGE client5a socket34 2550 8415-8670
START_RANGE client5a socket35 2550 8670-8925
START_RANGE client5a socket36 2550 8925-9180

START_RANGE client5a socket37 2550 9180-9435
START_RANGE client5a socket38 2550 9435-9690
START_RANGE client5a socket39 2550 9690-9945
START_RANGE client5a socket40 2550 9945-10200

START_RANGE client6a socket41 2550 10200-10455
START_RANGE client6a socket42 2550 10455-10710
START_RANGE client6a socket43 2550 10710-10965
START_RANGE client6a socket44 2550 10965-11220

START_RANGE client6a socket45 2550 11220-11475
START_RANGE client6a socket46 2550 11475-11730
START_RANGE client6a socket47 2550 11730-11985
START_RANGE client6a socket48 2550 11985-12240

START_RANGE client7a socket49 2550 12240-12495
START_RANGE client7a socket50 2550 12495-12750
START_RANGE client7a socket51 2550 12750-13005
START_RANGE client7a socket52 2550 13005-13260

START_RANGE client7a socket53 2550 13260-13515
START_RANGE client7a socket54 2550 13515-13770
START_RANGE client7a socket55 2550 13770-14025
START_RANGE client7a socket56 2550 14025-14280

```

START_RANGE client8a socket57 2550 14280-14535
START_RANGE client8a socket58 2550 14535-14790
START_RANGE client8a socket59 2550 14790-15045
START_RANGE client8a socket60 2550 15045-15300

START_RANGE client8a socket61 2550 15300-15555
START_RANGE client8a socket62 2550 15555-15810
START_RANGE client8a socket63 2550 15810-16065
START_RANGE client8a socket64 2550 16065-16320

#elif MASTER_NUM3

START_RANGE client9a socket65 2550 16320-16575
START_RANGE client9a socket66 2550 16575-16830
START_RANGE client9a socket67 2550 16830-17085
START_RANGE client9a socket68 2550 17085-17340

START_RANGE client9a socket69 2550 17340-17595
START_RANGE client9a socket70 2550 17595-17850
START_RANGE client9a socket71 2550 17850-18105
START_RANGE client9a socket72 2550 18105-18360

START_RANGE client10a socket73 2550 18360-18615
START_RANGE client10a socket74 2550 18615-18870
START_RANGE client10a socket75 2550 18870-19125
START_RANGE client10a socket76 2550 19125-19380

START_RANGE client10a socket77 2550 19380-19635
START_RANGE client10a socket78 2550 19635-19890
START_RANGE client10a socket79 2550 19890-20145
START_RANGE client10a socket80 2550 20145-20400

START_RANGE client11a socket81 2550 20400-20655
START_RANGE client11a socket82 2550 20655-20910
START_RANGE client11a socket83 2550 20910-21165
START_RANGE client11a socket84 2550 21165-21420

START_RANGE client11a socket85 2550 21420-21675
START_RANGE client11a socket86 2550 21675-21930
START_RANGE client11a socket87 2550 21930-22185
START_RANGE client11a socket88 2550 22185-22440

START_RANGE client12a socket89 2550 22440-22695
START_RANGE client12a socket90 2550 22695-22950
START_RANGE client12a socket91 2550 22950-23205
START_RANGE client12a socket92 2550 23205-23460

START_RANGE client12a socket93 2550 23460-23715
START_RANGE client12a socket94 2550 23715-23970
START_RANGE client12a socket95 2550 23970-24225
START_RANGE client12a socket96 2550 24225-24480

#elif MASTER_NUM4

START_RANGE client13a socket97 2550 24480-24735
START_RANGE client13a socket98 2550 24735-24990
START_RANGE client13a socket99 2550 24990-25245
START_RANGE client13a socket100 2550 25245-25500

START_RANGE client13a socket101 2550 25500-25755
START_RANGE client13a socket102 2550 25755-26010
START_RANGE client13a socket103 2550 26010-26265
START_RANGE client13a socket104 2550 26265-26520

START_RANGE client14a socket105 2550 26520-26775
START_RANGE client14a socket106 2550 26775-27030
START_RANGE client14a socket107 2550 27030-27285
START_RANGE client14a socket108 2550 27285-27540

START_RANGE client14a socket109 2550 27540-27795
START_RANGE client14a socket110 2550 27795-28050
START_RANGE client14a socket111 2550 28050-28305
START_RANGE client14a socket112 2550 28305-28560

START_RANGE client15a socket113 2550 28560-28815
START_RANGE client15a socket114 2550 28815-29070
START_RANGE client15a socket115 2550 29070-29325
START_RANGE client15a socket116 2550 29325-29580

START_RANGE client15a socket117 2550 29580-29835
START_RANGE client15a socket118 2550 29835-30090
START_RANGE client15a socket119 2550 30090-30345
START_RANGE client15a socket120 2550 30345-30600

START_RANGE client16a socket121 2550 30600-30855
START_RANGE client16a socket122 2550 30855-31110
START_RANGE client16a socket123 2550 31110-31365
START_RANGE client16a socket124 2550 31365-31620

START_RANGE client16a socket125 2550 31620-31875
START_RANGE client16a socket126 2550 31875-32130
START_RANGE client16a socket127 2550 32130-32385
START_RANGE client16a socket128 2550 32385-32640

#elif MASTER_NUM5

START_RANGE client17a socket129 2550 32640-32895
START_RANGE client17a socket130 2550 32895-33150
START_RANGE client17a socket131 2550 33150-33405
START_RANGE client17a socket132 2550 33405-33660

START_RANGE client17a socket133 2550 33660-33915
START_RANGE client17a socket134 2550 33915-34170
START_RANGE client17a socket135 2550 34170-34425
START_RANGE client17a socket136 2550 34425-34680

START_RANGE client18a socket137 2550 34680-34935
START_RANGE client18a socket138 2550 34935-35190
START_RANGE client18a socket139 2550 35190-35445
START_RANGE client18a socket140 2550 35445-35700

START_RANGE client18a socket141 2550 35700-35955
START_RANGE client18a socket142 2550 35955-36210
START_RANGE client18a socket143 2550 36210-36465
START_RANGE client18a socket144 2550 36465-36720

START_RANGE client19a socket145 2550 36720-36975
START_RANGE client19a socket146 2550 36975-37230
START_RANGE client19a socket147 2550 37230-37485
START_RANGE client19a socket148 2550 37485-37740

START_RANGE client19a socket149 2550 37740-37995
START_RANGE client19a socket150 2550 37995-38250
START_RANGE client19a socket151 2550 38250-38505
START_RANGE client19a socket152 2550 38505-38760

START_RANGE client20a socket153 2550 38760-39015
START_RANGE client20a socket154 2550 39015-39270
START_RANGE client20a socket155 2550 39270-39525
START_RANGE client20a socket156 2550 39525-39780

START_RANGE client20a socket157 2550 39780-40035
START_RANGE client20a socket158 2550 40035-40290
START_RANGE client20a socket159 2550 40290-40545
START_RANGE client20a socket160 2550 40545-40800

#elif MASTER_NUM6

START_RANGE client21a socket161 2550 40800-41055
START_RANGE client21a socket162 2550 41055-41310
START_RANGE client21a socket163 2550 41310-41565
START_RANGE client21a socket164 2550 41565-41820

START_RANGE client21a socket165 2550 41820-42075
START_RANGE client21a socket166 2550 42075-42330
START_RANGE client21a socket167 2550 42330-42585
START_RANGE client21a socket168 2550 42585-42840

START_RANGE client22a socket169 2550 42840-43095
START_RANGE client22a socket170 2550 43095-43350
START_RANGE client22a socket171 2550 43350-43605
START_RANGE client22a socket172 2550 43605-43860

START_RANGE client22a socket173 2550 43860-44115
START_RANGE client22a socket174 2550 44115-44370
START_RANGE client22a socket175 2550 44370-44625
START_RANGE client22a socket176 2550 44625-44880

START_RANGE client23a socket177 2550 44880-45135
START_RANGE client23a socket178 2550 45135-45390
START_RANGE client23a socket179 2550 45390-45645
START_RANGE client23a socket180 2550 45645-45900

START_RANGE client23a socket181 2550 45900-46155
START_RANGE client23a socket182 2550 46155-46410
START_RANGE client23a socket183 2550 46410-46665
START_RANGE client23a socket184 2550 46665-46920

START_RANGE client24a socket185 2550 46920-47175
START_RANGE client24a socket186 2550 47175-47430
START_RANGE client24a socket187 2550 47430-47685
START_RANGE client24a socket188 2550 47685-47940

START_RANGE client24a socket189 2550 47940-48195
START_RANGE client24a socket190 2550 48195-48450
START_RANGE client24a socket191 2550 48450-48705
START_RANGE client24a socket192 2550 48705-48960

#elif MASTER_NUM7

START_RANGE client25a socket193 2550 48960-49215
START_RANGE client25a socket194 2550 49215-49470
START_RANGE client25a socket195 2550 49470-49725
START_RANGE client25a socket196 2550 49725-49980

START_RANGE client25a socket197 2550 49980-50235
START_RANGE client25a socket198 2550 50235-50490
START_RANGE client25a socket199 2550 50490-50745
START_RANGE client25a socket200 2550 50745-51000

```
START_RANGE client26a socket201 2550 51000-51255
START_RANGE client26a socket202 2550 51255-51510
START_RANGE client26a socket203 2550 51510-51765
START_RANGE client26a socket204 2550 51765-52020
```

```
START_RANGE client26a socket205 2550 52020-52275
START_RANGE client26a socket206 2550 52275-52530
START_RANGE client26a socket207 2550 52530-52785
START_RANGE client26a socket208 2550 52785-53040
```

```
START_RANGE client27a socket209 2550 53040-53295
START_RANGE client27a socket210 2550 53295-53550
START_RANGE client27a socket211 2550 53550-53805
START_RANGE client27a socket212 2550 53805-54060
```

```
START_RANGE client27a socket213 2550 54060-54315
START_RANGE client27a socket214 2550 54315-54570
START_RANGE client27a socket215 2550 54570-54825
START_RANGE client27a socket216 2550 54825-55080
```

```
START_RANGE client28a socket217 2550 55080-55335
START_RANGE client28a socket218 2550 55335-55590
START_RANGE client28a socket219 2550 55590-55845
START_RANGE client28a socket220 2550 55845-56100
```

```
START_RANGE client28a socket221 2550 56100-56355
START_RANGE client28a socket222 2550 56355-56610
START_RANGE client28a socket223 2550 56610-56865
START_RANGE client28a socket224 2550 56865-57120
```

```
#elif MASTER_NUM8
START_RANGE client29a socket225 2550 57120-57375
START_RANGE client29a socket226 2550 57375-57630
START_RANGE client29a socket227 2550 57630-57885
START_RANGE client29a socket228 2550 57885-58140
```

```
START_RANGE client29a socket229 2550 58140-58395
START_RANGE client29a socket230 2550 58395-58650
START_RANGE client29a socket231 2550 58650-58905
START_RANGE client29a socket232 2550 58905-59160
```

```
START_RANGE client30a socket233 2550 59160-59415
START_RANGE client30a socket234 2550 59415-59670
START_RANGE client30a socket235 2550 59670-59925
START_RANGE client30a socket236 2550 59925-60180
```

```
START_RANGE client30a socket237 2550 60180-60435
START_RANGE client30a socket238 2550 60435-60690
START_RANGE client30a socket239 2550 60690-60945
START_RANGE client30a socket240 2550 60945-61200
```

```
START_RANGE client31a socket241 2550 61200-61455
START_RANGE client31a socket242 2550 61455-61710
START_RANGE client31a socket243 2550 61710-61965
START_RANGE client31a socket244 2550 61965-62220
```

```
START_RANGE client31a socket245 2550 62220-62475
START_RANGE client31a socket246 2550 62475-62730
START_RANGE client31a socket247 2550 62730-62985
START_RANGE client31a socket248 2550 62985-63240
```

```
START_RANGE client32a socket249 2550 63240-63495
```

```
START_RANGE client32a socket250 2550 63495-63750
START_RANGE client32a socket251 2550 63750-64005
START_RANGE client32a socket252 2550 64005-64260
```

```
START_RANGE client32a socket253 2550 64260-64515
START_RANGE client32a socket254 2550 64515-64770
START_RANGE client32a socket255 2550 64770-65025
START_RANGE client32a socket256 2550 65025-65280
```

```
#elif MASTER_NUM9
START_RANGE client33a socket257 2550 65280-65535
START_RANGE client33a socket258 2550 65535-65790
START_RANGE client33a socket259 2550 65790-66045
START_RANGE client33a socket260 2550 66045-66300
```

```
START_RANGE client33a socket261 2550 66300-66555
START_RANGE client33a socket262 2550 66555-66810
START_RANGE client33a socket263 2550 66810-67065
START_RANGE client33a socket264 2550 67065-67320
```

```
START_RANGE client34a socket265 2550 67320-67575
START_RANGE client34a socket266 2550 67575-67830
START_RANGE client34a socket267 2550 67830-68085
START_RANGE client34a socket268 2550 68085-68340
```

```
START_RANGE client34a socket269 2550 68340-68595
START_RANGE client34a socket270 2550 68595-68850
START_RANGE client34a socket271 2550 68850-69105
START_RANGE client34a socket272 2550 69105-69360
```

```
START_RANGE client35a socket273 2550 69360-69615
START_RANGE client35a socket274 2550 69615-69870
START_RANGE client35a socket275 2550 69870-70125
START_RANGE client35a socket276 2550 70125-70380
```

```
START_RANGE client35a socket277 2550 70380-70635
START_RANGE client35a socket278 2550 70635-70890
START_RANGE client35a socket279 2550 70890-71145
START_RANGE client35a socket280 2550 71145-71400
```

```
START_RANGE client36a socket281 2550 71400-71655
START_RANGE client36a socket282 2550 71655-71910
START_RANGE client36a socket283 2550 71910-72165
START_RANGE client36a socket284 2550 72165-72420
```

```
START_RANGE client36a socket285 2550 72420-72675
START_RANGE client36a socket286 2550 72675-72930
START_RANGE client36a socket287 2550 72930-73185
START_RANGE client36a socket288 2550 73185-73440
```

```
#elif MASTER_NUM10
START_RANGE client37a socket289 2550 73440-73695
START_RANGE client37a socket290 2550 73695-73950
START_RANGE client37a socket291 2550 73950-74205
START_RANGE client37a socket292 2550 74205-74460
```

```
START_RANGE client37a socket293 2550 74460-74715
START_RANGE client37a socket294 2550 74715-74970
START_RANGE client37a socket295 2550 74970-75225
START_RANGE client37a socket296 2550 75225-75480
```

```
START_RANGE client38a socket297 2550 75480-75735
```

```
START_RANGE client38a socket298 2550 75735-75990
START_RANGE client38a socket299 2550 75990-76245
START_RANGE client38a socket300 2550 76245-76500
```

```
START_RANGE client38a socket301 2550 76500-76755
START_RANGE client38a socket302 2550 76755-77010
START_RANGE client38a socket303 2550 77010-77265
START_RANGE client38a socket304 2550 77265-77520
```

```
START_RANGE client39a socket305 2550 77520-77775
START_RANGE client39a socket306 2550 77775-78030
START_RANGE client39a socket307 2550 78030-78285
START_RANGE client39a socket308 2550 78285-78540
```

```
START_RANGE client39a socket309 2550 78540-78795
START_RANGE client39a socket310 2550 78795-79050
START_RANGE client39a socket311 2550 79050-79305
START_RANGE client39a socket312 2550 79305-79560
```

```
START_RANGE client40a socket313 2550 79560-79815
START_RANGE client40a socket314 2550 79815-80070
START_RANGE client40a socket315 2550 80070-80325
START_RANGE client40a socket316 2550 80325-80580
```

```
START_RANGE client40a socket317 2550 80580-80835
START_RANGE client40a socket318 2550 80835-81090
START_RANGE client40a socket319 2550 81090-81345
START_RANGE client40a socket320 2550 81345-81600
```

```
#elif MASTER_NUM11
#endif
/*-----*/
#define TES_FLAG_TRACE 0x00000010
#define TES_FLAG_KEYSTROKE_TIME 0x00000200
#define TES_FLAG_LOCAL_LOG 0x00000400
#define TES_FLAG_LOCAL_TRACE 0x00000800
#define TES_FLAG_LOCAL_IPRINT 0x00004000
#if 0
/* SETFLAG ALL TES_FLAG_TRACE */
SETFLAG ALL TES_FLAG_LOCAL_TRACE
SETFLAG ALL TES_FLAG_LOCAL_IPRINT
#endif
#if 0
SETFLAG client31 telnet 1 TES_FLAG_KEYSTROKE_TIME
#endif
```

D.2 RTE Scripts

tpccWeb.h

```
/*
*****
** Project : AIX DB/2 TPC-C
** Component : TPC-C/Client
** Name : tpccWeb.h
** Title : rte web defines
*****
** Copyright (c) IBM US - AUSTIN 2000
```



```

extern SlaveStatus slave_status[MAX_SLAVES];
extern Stats status[MAX_TRAN_TYPE][MAX_TIMES];
extern WINDOW *statistics_win;
extern UserGlobal *shmglobal;
/* Transaction mix parameters */
double ratio_desired[6], ratio_min[6], ratio_max[6],
ratio_range[6];
char *ratio_names[] = { "RTE", "NEWORDER", "PAYMENT",
"ORDSTAT", "DELIVERY",
"STOCKLEV", NULL };
char *Status_Names[] = {"Menu", "Keying", "Response",
"Think"};

char *transaction_names[] = { "RTE", "New Order", "Payment",
"Order Stat",
"Delivery", "Stock Level", NULL };
static int current_status = 2, status_needs_refresh = 1;
int user_statistics_print(void) {
    int i;
    static int count = 0;
    double ratios[6];
    if (status_needs_refresh) {
        count = 0;
        status_needs_refresh = 0;
        wmove (statistics_win, 0, 0);
        wprintw (statistics_win, "%11s %8s %8s %8s %8s %8s
%6s %6s %6s",
                Status_Names[current_status], "90%", "Avg", "Min",
"Max",
                "Samples", "Ratio", "Mix", "Think");
    }
    make_ratios(ratios);
    for (i = 1; i <= 5; i++) {
        if (count % 10 == 0) {
            wmove (statistics_win, i, 0);
            wprintw (statistics_win, "%11s %8.2f",
                    transaction_names[i],
                    status[i][current_status].ninety()/1000.0);
            count = 0;
        }
        wmove (statistics_win, i, 21);
        wprintw (statistics_win, "%8.2f %8.2f %8.2f %8d %6.2f
%6.2f %6.2f",
                status[i][current_status].average()/1000.0,
                status[i][current_status].min()/1000.0,
                status[i][current_status].max()/1000.0,
                status[i][current_status].samples(),
                ratios[i], shmglobal->chances[i],
                status[i][3].average()/1000.0);
    }
    wmove (statistics_win, 7, 0);
    extern int runtime_counts[MAX_TRAN_TYPE];
    extern int begin_time, ramp_up, run_time;
    int start = interval_start_time;
    int stop = interval_stop_time;
    double interval = ((double)(stop-start) / (1000*60));
    double samples = status[1][2].samples();
    if (interval <= 0 || samples <= 0) {
        wprintw (statistics_win, "TPM-C: %7s / ", "-----");
    } else {
        wprintw (statistics_win, "TPM-C: %7.2f / ",
                samples/interval);
    }
}

```

```

}
samples = runtime_counts[1];
if (samples > 0) {
    start = begin_time + ((ramp_up >= 0) ? ramp_up : 0);
    if (run_time > 0 && stop > begin_time + ramp_up +
run_time) {
        stop = begin_time + ramp_up + run_time;
    }
    interval = (double)(stop - start)/(1000.0*60.0);
    wprintw (statistics_win, "%7.2f", samples/interval);
} else {
    wprintw (statistics_win, "-----");
}
count++;
return RTE_OK;
}
extern int login_begin;
int login_max_load;
#ifdef WHSEARRAYDBG
int outofboundwarn;
#endif
extern int min_warehouse;
extern int max_warehouse;
const int MAX_WAREHOUSES=100000;
/* All of this 10 stuff is district size. Should be a constant.
Maybe fix that later */
int num_warehouses = -1;
int warehouses[MAX_WAREHOUSES*10];
int user_spawn(int min, int max, int number, int *length, char
*buffer) {
    //int user_spawn(int number, int *length, char *buffer) {
    int i, min_index;
    int adj_wh = num_warehouses; // adjusted warehouse
number
    UserSpawnData *ptr = (UserSpawnData *)buffer;
    *length = sizeof(*ptr);
    // min_index = 0;
    // for (i = 1; i < (num_warehouses)*10 && i <
MAX_WAREHOUSES*10; i++) {
    //
    // if both min and max are zero, running START, otherwise
running
    // START_RANGE. Must also determine what the ending
warehouse number
    // will be for said range
    //
    if (min == 0 && max == 0) {
        min++;
        min_index = 0;
    } else {
        adj_wh = max; // inclusive range of wh-s
min = min * 10;
        min_index = min;
    }
    for (i = min ; i < (adj_wh)*10 && i <
((MAX_WAREHOUSES+min_warehouse)*10); i++) {
        if (warehouses[i - (min_warehouse*10)] <
warehouses[min_index - (min_warehouse*10)]) {
            min_index = i;
        }
    }
}
}

```

```

ptr->Warehouse = min_index / 10 + 1;
ptr->District = min_index % 10 + 1;
#ifdef WHSEARRAYDBG
if ((min_index - (min_warehouse*10) < 0) || (min_index -
(min_warehouse*10) >= (MAX_WAREHOUSES*10))) {
    if (outofboundwarn) {
        iprint (IPRINT_INFO, "(spawn) Out of range warehouse
number %d, (%d-%d (start) = %d (rel. num)\n",
                min_index, min_index, min_warehouse, min_index -
(min_warehouse*10));
        outofboundwarn=0;
    }
}
#endif
warehouses[min_index - (min_warehouse*10)]++;
/* iprint (IPRINT_INFO, "Driver for Warehouse %d, District
%d started. warehouses[%d]++ = %d\n",
        ptr->Warehouse, ptr->District, min_index,
        warehouses[min_index - (min_warehouse*10)]); */
return RTE_OK;
}
int user_finished(int length, char *buffer) {
    UserSpawnData *ptr = (UserSpawnData *)buffer;
    int temp = (ptr->Warehouse-1)*10+ptr->District-1;
#ifdef WHSEARRAYDBG
if ((temp - min_warehouse*10 < 0) || (temp -
min_warehouse*10 >= MAX_WAREHOUSES*10)) {
    if (outofboundwarn) {
        iprint (IPRINT_INFO, "(finish) Out of range warehouse
number %d, (%d-%d (start) = %d (rel. num)\n",
                min_index, min_index, min_warehouse, min_index -
(min_warehouse*10));
        outofboundwarn=0;
    }
}
#endif
warehouses[temp - (min_warehouse*10)]--;
/* iprint (IPRINT_INFO, "Driver for Warehouse %d, District
%d died. warehouses[%d]-- = %d\n",
        ptr->Warehouse, ptr->District, temp, warehouses[temp -
(min_warehouse*10)]); */
return RTE_OK;
}
double limit(double min, double max, double val) {
    if (val < min)
        return min;
    if (val > max)
        return max;
    return val;
}
int make_ratios (double *buffer) {
    int neword = status[NEWORDER][0].samples();
    int payment = status[PAYMENT][0].samples();
    int ordstat = status[ORDSTAT][0].samples();
    int delivery = status[DELIVERY][0].samples();
    int stocklev = status[STOCKLEV][0].samples();
    int total = neword + payment + ordstat + delivery + stocklev;
    int i;
    if (total == 0) {
        buffer[NEWORDER] = 100.0;
        for (i = 2; i < 6; i++) {
            buffer[i] = ratio_desired[i];
        }
    }
}

```

```

        buffer[NEWORDER] -= buffer[i];
    }
    return 0;
}
buffer[PAYMENT] = (double)payment / (double)total * 100.0;
buffer[ORDSTAT] = (double)ordstat / (double)total * 100.0;
buffer[DELIVERY] = (double)delivery / (double)total * 100.0;
buffer[STOCKLEV] = (double)stocklev / (double)total * 100.0;
buffer[NEWORDER] = 100.0- buffer[PAYMENT] -
buffer[ORDSTAT] -
        buffer[DELIVERY] - buffer[STOCKLEV];
return total;
}
int user_global_update(int *length, char *buffer) {
    UserGlobal *shmglobal = (UserGlobal *)buffer;
    static double last[6];
    static last_test_state = 0;
    static int users_last=1;
    double ratios[6];
    double current[6];
    int i, different = 0;
    int desired = 0;
    int host_busy, all_zero;
    *length = sizeof(*shmglobal);
    make_ratios(ratios);
    /* Calculate ratios we want for next time */
    if (ramp_up_complete) {
        current[NEWORDER] = 100.0;
        for (i = 2; i < 6; i++) {
            if (ratio_desired[i] > ratios[i]) {
                current[i] = ratio_max[i];
            } else {
                current[i] = 2*ratio_desired[i] - ratios[i];
                if (current[i] < ratio_min[i])
                    current[i] = ratio_min[i];
            }
            current[NEWORDER] -= current[i];
        }
    } else {
        for (i = 1; i < 6; i++) {
            current[i] = ratio_desired[i];
        }
    }
    /* Add up all the users */
    /* This needs to be changed to be more transparent */
    shmglobal->total_users = 0;
    for (i = 0; i < MAX_SLAVES; i++) {
        shmglobal->total_users += slave_status[i].active;
        desired += slave_status[i].desired;
    }
    /* Count up number of warehouses we WANT to have */
    if (num_warehouses < 0) {
        num_warehouses = (desired-1)/10+1;
    }
    shmglobal->max_warehouses = num_warehouses;
    host_busy = 0;
    all_zero = 1;
    for (i = 1; i <= 5; i++) {
        if (status[i][current_status].average() != 0) {
            all_zero = 0;
        }
    }
}

```

```

        if ( status[i][current_status].average()/1000.0 >
login_max_load ) {
            host_busy = 1;
        }
    }
    if (shmglobal->host_busy && all_zero) {
        host_busy = 1;
    }
    if (host_busy != shmglobal->host_busy) {
        shmglobal->host_busy = host_busy;
        different = 1;
    }
    for (i = 2; i < 6; i++) {
        if (current[i] != last[i])
            different = 1;
    }
    if (last_test_state != shmglobal->test_state) {
        different = 1;
        last_test_state = shmglobal->test_state;
    }
    /* Don't send if it's the same as last time
    if ( !different && shmglobal->total_users == users_last ) {
        return RTE_ERROR;
    }
    users_last = shmglobal->total_users;
    for (i = 1; i < 6; i++) {
        shmglobal->chances[i] = last[i] = current[i];
    }
    return RTE_OK;
}
int user_isbusy() {
    return shmglobal->host_busy;
}

int parse_array(char *string, int max, int *buffer) {
    int i, rc;
    char *ptr;
    char *temp = strdup(string);
    ptr = strtok(temp, ",");
    for (i = 0; ptr && i < max; i++) {
        rc = sscanf(ptr, "%d", &buffer[i]);
        if (rc < 1) {
            free(temp);
            return i;
        }
        ptr = strtok(NULL, ",");
    }
    free(temp);
    return i;
}

int parse_array(char *string, int max, double *buffer) {
    int i, rc;
    char *ptr;
    char *temp = strdup(string);
    ptr = strtok(temp, ",");
    for (i = 0; ptr && i < max; i++) {
        rc = sscanf(ptr, "%lf", &buffer[i]);
        if (rc < 1) {
            free(temp);
            return i;
        }
        ptr = strtok(NULL, ",");
    }
}

```

```

    }
    free(temp);
    return i;
}

int user_init() {
    double dbuffer[32];
    int rc, i;
    char *ptr;
    if (get_variable("KEYSTROKE_SLEEP", &shmglobal-
>keystroke_sleep) != RTE_OK) {
        shmglobal->keystroke_sleep = 0;
    }
    if (get_variable("LOGIN_TIMEOUT", &shmglobal-
>login_timeout) != RTE_OK) {
        shmglobal->login_timeout = 120; /* 2 minutes */
    }
    if (get_variable("KEYSTROKE_PACKET_SIZE", &shmglobal-
>keystroke_packet_size) != RTE_OK) {
        shmglobal->keystroke_packet_size = 0;
    }
    shmglobal->login_timeout *= 1000;
    if (get_variable("LOGIN_MAX_LOAD", &login_max_load) !=
RTE_OK) {
        login_max_load = 2;
    }
    if (get_variable("WAREHOUSES", &num_warehouses) !=
RTE_OK) {
        num_warehouses = -1;
    }
    if (get_variable("LASTC", &shmglobal->lastc) != RTE_OK) {
        shmglobal->lastc = 193; /* 2 minutes */
    }
    iprint(IPRINT_INFO, "Login Timeout = %s\n",
        mstoa_withfrac(shmglobal->login_timeout, 0));
    iprint(IPRINT_INFO, "Keystroke Sleep = %s\n",
        mstoa_withfrac(shmglobal->keystroke_sleep*1000, 0));
    iprint(IPRINT_INFO, "Keystroke Packet Size= %d\n",
shmglobal->keystroke_packet_size);
    if (num_warehouses >= 0) {
        iprint(IPRINT_INFO, "Fixed Warehouses to = %d\n",
num_warehouses);
    }
    if (!(ptr = get_variable("NEWORDER"))) {
        iprint_error ("Error. NEWORDER variable not found\n");
        exit (1);
    }
    if (parse_array(ptr, 3, dbuffer)!=3) {
        iprint_error ("Error. NEWORDER should be think,
emulex_menu, emulex_response");
        exit (1);
    }
    shmglobal->think [NEWORDER] = dbuffer[0];
    shmglobal->emulex_menu [NEWORDER] = dbuffer[1];
    shmglobal->emulex_response[NEWORDER] = dbuffer[2];
    shmglobal->test_state = 0;
    for (i = 2; i < 6; i++) {
        if (!(ptr = get_variable(ratio_names[i])) ||
(parse_array(ptr, 6, dbuffer)!=6)) {
            iprint(__FILE__, __LINE__, IPRINT_ERROR,
                "Error. %s should be think, emulex_menu,
emulex_response, desired, min, max",

```



```

        ratio_names[i]);
    exit (1);
}
shmglobal->think[i] = dbuffer[0];
shmglobal->emulex_menu[i] = dbuffer[1];
shmglobal->emulex_response[i] = dbuffer[2];
ratio_desired[i] = dbuffer[3];
ratio_min[i] = dbuffer[4];
ratio_max[i] = dbuffer[5];
ratio_range[i] = ratio_max[i]-ratio_min[i];
}
for (i=0; i < (MAX_WAREHOUSES*10); i++) {
    warehouses[i] = 0;
}
#ifdef WHSEARRAYDBG
    outofboundwarn=1;
#endif
return RTE_OK;
}
int user_extra_data(header_s *header) {
    int i;
    int num_timestamps;
    if (header->data_type != RTE_ITEM_KEYSTROKE_TIMES)
        return RTE_OK;
    int *times = (int *)((char *)header+sizeof(struct header_s));
    num_timestamps = header->user_data_length / 4 - 1;
    iprint (IPRINT_TRACE, "Keystroke times = ");
    for (i = 0 ; i < num_timestamps; i++) {
        iprint (IPRINT_TRACE, "%d ", times[i]);
    }
    iprint (IPRINT_TRACE, "\n", times[i]);

    return RTE_OK;
}
int user_process_command(char *command) {
    char buffer[256], *ptr;
    int i, found, len;
    strncpy (buffer, command, 256);
    ptr = strtok (buffer, " \t");
    found = 0;
    printf ("user_process_command('%s')\n",ptr);
    if (!strcasecmp (ptr, "pause")) {
        shmglobal->test_state = 1;
    } else if (!strcasecmp (ptr, "warmup")) {
        shmglobal->test_state = 2;
    } else if (!strcasecmp (ptr, "notest")) {
        shmglobal->test_state = 0;
    } else if (!strcasecmp (ptr, "login_max_load?")) {
        iprint (IPRINT_WARNING, "Current LOGIN_MAX_LOAD
= %d\n", login_max_load);
    } else if (!strcasecmp (command, "login_max_load=",15)) {
        login_max_load=atoi(command+15);
        iprint (IPRINT_WARNING, "Set LOGIN_MAX_LOAD =
%d\n", login_max_load);
    } else if (!strcasecmp (ptr, "display")) {
        while (ptr && (ptr = strtok(NULL, " \t"))) {
            if (*ptr == '\0')
                continue;
            for (i = 0; i < 5; i++) {
                len = min(strlen(Status_Names[i]), strlen(ptr));
                if (!strcasecmp (ptr, Status_Names[i], len)) {
                    status_needs_refresh = found = 1;

```

```

        current_status = i;
        return RTE_OK;
    }
}
iprint (IPRINT_WARNING, "Unknown type to display:
%s\n", ptr);
}
} else {
    iprint (IPRINT_WARNING, "Unknown Command: '%s'\n",
command);
    return RTE_ERROR;
}
}
return RTE_OK;
}
int transaction_process () {
    return RTE_OK;
}
int user_begin() {
    return RTE_OK;
}
}
/*
void user_make_header(char *buffer) {
    int i;
    struct user_data_header *data = (struct user_data_header
*)buffer;
}
*/

```

user_slave.C

```

/*****
*****/
/* user_slave.C Audit: 05/30/96 */
/*****
*****/
static char *rcsid="$Id: user_slave.C,v 1.1 1999/02/22 06:31:06
channui Exp $";
/*****
*****/
/*
*** TPCC FILE FOR ALL USERS
***
*****/
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/time.h>
#include "rte_slave.h"
#include "user_tpcc.h"
/* This MUST match the corresponding one in client's inout.h
file! */
#define TRIGGER "\021"
//define NOSLEEP
// Increased EXPECT_TIMEOUT from 600000 - oz 10/20/97
#define EXPECT_TIMEOUT 6000000
#define KEYWAIT_FUDGE 5000
extern SHM_Slave *shm;
extern TableEntrySlave *shmentry;
extern DriverStatus *status;
extern echo_trace(char *);

```

```

extern echo_trace();
extern char *expect_save;
extern char *expect_buffer_return();
const char *SQL_TPERRNO_MESSAGE = "tperrno";
const char *SQL_RTN_MESSAGE = "rtn:";
const char *SQL_FATAL_MESSAGE = "SQL Fatal Error";
const char *ROLLBACK_MESSAGE = "Item number
is not valid";
const char *CUSTOMER_ID_STRING = "Customer: ";
int WHSEID; /* warehouse number for each
users */
/*****
*****/
/* The "uniform()" function has range of the absolute value of the
*/
/* difference between the min. and the max values upto
2147483647. */
/*****
*****/
/*-----*/
/* NURand */
/*-----*/
/* A: 255 for C_LAST, 1023 for C_ID, 8191 for OL_I_ID */
/* x: 0 for C_LAST, 1 for C_ID and OL_I_ID */
/* y: 999 for C_LAST, 3000 for C_ID, 100000 for OL_I_ID */
/*-----*/
long
NURand(int A, int x, int y, long cval)
{
    return (((long) uniform((long) 0, (long) A) | (long)
uniform((long) x, (long) y)) + cval) % (y - x + 1) + x;
}
/*-----*/
/* getname */
/*-----*/
/* generates a random number from 0 to 999 inclusive */
/* a random name is generated by associating a random */
/* string with each digit of the generated number */
/* three strings are concatenated to generate lastname */
/*-----*/
char *
getname()
{
    char *last_name_parts[] =
    {
        "BAR",
        "OUGHT",
        "ABLE",
        "PRI",
        "PRES",
        "ESE",
        "ANTI",
        "CALLY",
        "ATION",
        "EING"
    };
    static char lastname[128];
    int random_num;
    #if 1
        random_num = NURand(255, 0, 999, shmglobal->lastc);
    #else
        random_num = NURand(255, 0, 999, LASTC);

```

```

#endif
strcpy(lastname, last_name_parts[random_num / 100]);
random_num %= 100;
strcat(lastname, last_name_parts[random_num / 10]);
random_num %= 10;
strcat(lastname, last_name_parts[random_num]);
return (lastname);
}
typedef struct gen_tran_s {
int invalid;
void *data;
long len;
long keywait;
long type;
char *menu;
char *results_request;
char *form_request;
} gen_tran_t;
typedef struct gen_tran_url_s
{
char *txn_form_url;
char *txn_results_url;
} gen_tran_url_s;
int generic_transaction( gen_tran_t *data, char *host)
{
char buffer[2048];
static int consecutive_errs = 0;
int rc;
set_typing_delay(0);
//iprint(IPRINT_TRACE, "> generic_transaction sleep (%d)
type(%d) *data (%d)\n", data->type, data->menu, data);
#ifdef NOSLEEP
if (shmglobal->test_state == 0)
transaction_sleep_do();
#endif
#ifdef EXPECT_TIMEOUT
int timeout = EXPECT_TIMEOUT;
#else
int timeout = 0;
#endif
// Start the transaction (MENU)
//iprint(IPRINT_TRACE, "> generic_transaction start (%d)\n",
data->type);
transaction_start(data->type, data->len, data->data);
//send menu request page
//iprint(IPRINT_TRACE, "> transmit data->menu: (%s)\n
request :(%s)", data->menu, data->form_request);
//iprint(IPRINT_TRACE, "> transmit data->menu: (%s)\n
request :(%s)", data->menu);
transmit(data->form_request);

echo_trace ("Waiting for Menu");
switch (*data->menu)
{
case '1':
rc =
expect_html(NORD_TITLE, timeout, NORD_TITLE_LEN);
break;
case '2':
rc =
expect_html(PYMT_TITLE, timeout, PYMT_TITLE_LEN);
break;

```

```

case '3':
rc =
expect_html(ORDS_TITLE, timeout, ORDS_TITLE_LEN);
break;
case '4':
rc =
expect_html(DLVY_TITLE, timeout, DLVY_TITLE_LEN);
break;
case '5':
rc =
expect_html(STOK_TITLE, timeout, STOK_TITLE_LEN);
break;
default:
rc = ERROR;
}
if(rc == ERROR)
{
iprint (IPRINT_ERROR, "Slave %d:Failed to receive
%s input form\n**Request-->%s<--\n",
shmentry->num, data->menu, data-
>form_request);
return (ERROR);
}
//if (expect_html(TRIGGER, timeout) == ERROR)
//{
// iprint (IPRINT_ERROR, "Slave %d: Failed to receive %s
screen\n",
// shmentry->num, data->menu);
// return (ERROR);
//}
#ifdef NOSLEEP
usleep(shmglobal->emulex_menu[data-
>type]*1000000.0+0.9);
#endif
// Send our request (KEYING)
transaction_mark(WHERE_NOW);
echo_trace ("Keying");
#ifdef NOSLEEP
usleep(data->keywait*1000000+KEYWAIT_FUDGE); //
Keying delay
#endif
// Wait for response (RESPONSE)
transaction_mark(WHERE_NOW);
//iprint(IPRINT_TRACE, "> transmit request :(%s)\n", data-
>results_request);
transmit(data->results_request);
echo_trace ("Wait for Response");
switch (*data->menu)
{
case '1':
rc =
expect_html(NORD_RESULTS_TITLE, timeout, NORD_RESULT
S_TITLE_LEN);
break;
case '2':
rc =
expect_html(PYMT_RESULTS_TITLE, timeout, PYMT_RESULT
S_TITLE_LEN);
break;
case '3':

```

```

rc =
expect_html(ORDS_RESULTS_TITLE, timeout, ORDS_RESULT
S_TITLE_LEN);
break;
case '4':
rc =
expect_html(DLVY_RESULTS_TITLE, timeout, DLVY_RESULTS
_TITLE_LEN);
break;
case '5':
rc =
expect_html(STOK_RESULTS_TITLE, timeout, STOK_RESULT
S_TITLE_LEN);
break;
default:
rc = ERROR;
}
if(rc == ERROR)
{
iprint (IPRINT_ERROR, "Slave %d:Failed to receive
%s result page\n**Request-->%s<--\n",
shmentry->num, data->menu, data-
>form_request);
return (ERROR);
}
// if (expect_html(TRIGGER, timeout) == ERROR) {
//iprint (IPRINT_ERROR, "Slave %d: Failed to receive %s
response\n",
// shmentry->num, data->menu);
// return (ERROR);
//}
#ifdef NOSLEEP
usleep(shmglobal->emulex_response[data-
>type]*1000000.0+0.9);
#endif
// Look for errors and set our think time (THINK)
transaction_mark(WHERE_NOW);
if (expect_buffer_search("ERROR:", 6))
{
FILE *fd;
if ((fd = fopen("/u/rte/error.txt", "a"))!=NULL)
{
fprintf(fd, "Slave %d: %s found '%s'\n%s\n",
shmentry->num, data->menu,
"ERROR:", expect_buffer_return());
fclose(fd);
}
data->invalid = 1;
iprint (IPRINT_ERROR, "Slave %d: %s found
%s'\n%s'\n",
shmentry->num, data->menu,
"ERROR:", expect_buffer_return());
// Very dangerous, keep going rather than exiting...
//return RTE_ERROR;
// Check for consecutive errors and if there are more
than
// 4 of them exit - allow for transient errors to make
// tuning and testing easier -oz
// In either case the transaction is marked as invalid and
// will be reported as an error by the analyze program.
if (consecutive_errs++ > 4)

```

```

        return RTE_ERROR;
    }
    else
    {
        consecutive_errs = 0;
    }
    echo_trace ("Thinking");
    transaction_sleep_set(neg_exp_4(shmglobal->think[data-
>type])*1000.0);
    //iprint(IPRINT_TRACE, "< generic_transaction finish\n");
    return (RTE_OK);
}
/*****
**/
/****          Delivery Transaction          ****/
/*****
**/
int
Delivery(char *host,int terminal)
{
    static struct delivery_struct delivery, delivery_new;
    int      rc;
    char     *ptr;
    gen_tran_t tran;
    tran.invalid = 0;
    tran.data = &delivery;
    tran.len = sizeof(delivery);
    tran.keywait = 2;
    tran.type = DELIVERY;
    tran.menu = "4";
    char dlv_url[128];
    char form_buffer[256];
    char results_buffer[256];
    tran.form_request = form_buffer;
    tran.results_request = results_buffer;
    //create dlv form request
    sprintf(dlv_url,DELIVERY_FORM_URL,terminal);
    sprintf(form_buffer,GET_REQUEST,dlv_url,host);
    // Set up all data for new transactions
    delivery_new.carrier = uniform(1, 10); // carrier # 1 to 10
    //create dlv results request
    sprintf(dlv_url,DELIVERY_RESULTS_URL,terminal,deliv
ery_new.carrier);
    sprintf(results_buffer,GET_REQUEST,dlv_url,host);

    // Go do the transaction
    rc = generic_transaction(&tran,host);
    delivery = delivery_new;
    delivery.invalid = tran.invalid;
    //iprint(IPRINT_TRACE,"dlv txn finished, rc:%d
tran.invalid:%d\n",rc,delivery.invalid);
    return (rc);
}
/*****
**/
/****          New Order Transaction          ****/
/*****
**/
int NewOrder(char *host,int terminal)
{
    static struct neword_struct neword, neword_new;

```

```

int      i, rc, whses, low_whse=1;
char nord_form_url[128];
char form_buffer[512];
char nord_results_url[2048];
char results_buffer[4096];
char     *ptr;
char     *ptr2;
const char *err_ptr;
gen_tran_t tran;
tran.invalid = 0;
tran.data = &neword;
tran.len = sizeof(neword);
tran.keywait = 18;
tran.type = NEWORDER;
tran.menu = "1";

tran.form_request = form_buffer;
tran.results_request = results_buffer;

neword_new.rollback=0;
/**** SECTION TO DETERMINE ROLLBACK TRANSACTION
FOR 1% OF NEW ORDERS ****/
neword_new.did = uniform(1, 10);
// district number
neword_new.cid = NURand(1023, 1, 3000,
CUSTC); // customer # 1 to 3000
neword_new.nloop = uniform(5, 15);
neword_new.olremote = 0;
whses = shmglobal->max_warehouses;
for (i = 0; i < neword_new.nloop; i++)
    {
        // Warehouse Number
        neword_new.item[i].olswid = WHSEID;
        if (whses > 1 && (uniform(0.0, 100.0) < 1.0))
            {
                /* for 1% of items (if * uniform()==0) */
                /* Generate a uniform whse number that's different
from WHSEID */
                neword_new.item[i].olswid =
(long) uniform((long) low_whse, (long)whses-
1);
                if (neword_new.item[i].olswid >= WHSEID)
                    neword_new.item[i].olswid++;
                neword_new.olremote++; // find total number of
remote order-lines
            }
            // Item number 1-100000
            neword_new.item[i].oliid = NURand(8191, 1,
100000, ITEM);
            // Quantity 1-10
            neword_new.item[i].olquantity = uniform(1, 10);
            /* end of for n_loop */
            // We occasionally force a transaction to have invalid data to
force a
            // rollback
            if (uniform(1, 5000) <= 50)
                neword_new.item[neword_new.nloop-1].oliid =
999999;
            neword_new.olremote = (neword_new.olremote > 0);
            //create new order form request
            sprintf(nord_form_url,NEW_ORDER_FORM_URL,termina
l);

```

```

//create get form request
sprintf(form_buffer,GET_REQUEST,nord_form_url,host);
//create new order results url
char itemString[1024];
ptr2=itemString;
short item_cmd_start = ITEM_CMD_ID_START;
for (i = 0; i < neword_new.nloop; i++)
    {
        ptr2 += sprintf(ptr2, NEW_ORDER_ITEM,
                        item_cmd_start++,
                        neword_new.item[i].olswid);

        ptr2 += sprintf(ptr2,NEW_ORDER_ITEM,
                        item_cmd_start++,
                        neword_new.item[i].oliid);

        ptr2 += sprintf(ptr2, NEW_ORDER_ITEM,
                        item_cmd_start++,
                        neword_new.item[i].olquantity);
    }
//seal up url w/ empty items
for (i = item_cmd_start;i <= ITEM_CMD_ID_END; i++)
    {
        ptr2 += sprintf(ptr2,NEW_ORDER_EMPTY_ITEM,i);
    }
// number of items to order (5-15)
//printf("total results url:NEW_ORDER_RESULTS_URL,te
rminal,
neword_new.did,neword_new.cid,
                        itemString);

//create get results request
sprintf(results_buffer,GET_REQUEST,nord_results_url,ho
st);

// Go do the transaction
rc = generic_transaction(&tran,host);
neword = neword_new;
neword.invalid = tran.invalid;
// Check for a rollback
if ((err_ptr = expect_buffer_search("Item number is not
valid",24)))
    {
        neword.rollback=1;
        echo_trace ("Found rollback!\n");
    }
// Grab the orderID from the
if (!(err_ptr = expect_buffer_search("Order Number: ",14)))
    {
        echo_trace ("Didn't find order-id for neworder");
        iprint (IPRINT_ERROR, "Neworder didn't have
Order-ID\n%s\n",expect_buffer_return());
        //iprint (IPRINT_ERROR, "Neworder didn't have
Order-ID\n");
        neword.oid = -1;
    }
    else
    {
        neword.oid = atoi(err_ptr+14);
    }

```

```

// iprint(IPRINT_ERROR,"New order order
id:%d\n",neword.oid);
}

// This is really not useful since we aren't going to be sending
individual
// keystrokes anymore
if (shmentry->flags & TES_FLAG_KEYSTROKE_TIME) {
log_data(RTE_ITEM_KEYSTROKE_TIMES,
keystroke_length*sizeof(int),keystroke_times);
}
//iprint(IPRINT_TRACE,"nord txn finished, rc:%d
tran.invalid:%d\n",rc,tran.invalid);
return (rc);
}

/*****
**/
/**      Order Status Transaction      **/
/*****
**/
int OrderStatus(char *host,int terminal) {
static struct ordstat_struct ordstat, ordstat_new;
//char  buffer[2048];
int rc;
char *ptr;
gen_tran_t tran;
tran.invalid = 0;
tran.data = &ordstat;
tran.len = sizeof(ordstat);
tran.keywait = 2;
tran.type = ORDSTAT;
tran.menu = "3";
//tran.request = buffer;

//Joe N.
char ords_url[256];
char form_buffer[512];
char results_buffer[2048];
tran.results_request = results_buffer;
tran.form_request = form_buffer;

//create order status form request
sprintf(ords_url,ORDER_STATUS_FORM_URL,terminal);
sprintf(form_buffer,GET_REQUEST,ords_url,host);

// Set up all data for new transactions
ordstat_new.did = uniform(1, 10); /* district number 1 to
10 */
if (uniform(1, 100) <= 60)
{
/* for 60% of transactions */
char *tmp = getname();
strcpy(ordstat_new.clast, tmp); /* by customer
last name */
if (ordstat_new.clast[0] < 'A' || ordstat_new.clast[0] >
'Z')
{
iprint (IPRINT_ERROR,
"ASSERTION: OrderStatus getname() returns invalid
name! '%s'\n",

```

```

ordstat_new.clast);
return RTE_ERROR;
}
ordstat_new.byname = 1;
ordstat_new.cid = 0;
}
else
{
ordstat_new.cid = NURand(1023, 1, 3000, CUSTC);
/* cust. # 1 to 3000 */
ordstat_new.byname = 0;
ordstat_new.clast[0] = (char) NULL;
}
//iprint(IPRINT_TRACE,"Order status fields,w_id:%d
d_id:%d n_d_id:%d c_id:%d\n", data->Warehouse,data-
>District,ordstat_new.did,ordstat_new.cid);
//create order status url request
if (ordstat_new.byname)

sprintf(ords_url,ORDER_STATUS_RESULTS_CLAST_U
RL,terminal,ordstat_new.did,

else

sprintf(ords_url,ORDER_STATUS_RESULTS_CID_URL,t
erminal,ordstat_new.did,

sprintf(results_buffer,GET_REQUEST,ords_url,host);

// Go do the transaction
rc = generic_transaction(&tran,host);
ordstat = ordstat_new;
ordstat.invalid = tran.invalid;
//iprint(IPRINT_TRACE,"ords txn finished, rc:%d
tran.invalid:%d\n",rc,tran.invalid);
return (rc);
}
/*****
**/
/**      Payment Transaction      **/
/*****
**/
int
Payment(char *host,int terminal)
{
static struct payment_struct payment, payment_new;
int dollars, cents, rc, whses, low_whse = 1;
char *ptr;
gen_tran_t tran;
tran.invalid = 0;
tran.data = &payment;
tran.len = sizeof(payment);
tran.keywait = 3;
tran.type = PAYMENT;
tran.menu = "2";

char pymt_url[128];
char form_buffer[256];
char results_buffer[2048];
tran.results_request = results_buffer;
tran.form_request = form_buffer;
//create pymt form url

```

```

sprintf(pymt_url,PAYMENT_FORM_URL,terminal);
sprintf(form_buffer,GET_REQUEST,pymt_url,host);

payment_new.did = uniform(1, 10); /* district number 1 to
10 */
if (uniform(1, 100) <= 60) /* for 60% of transactions */
strcpy(payment_new.clast, getname(), 17); /* by
customer last name
if (payment_new.clast[0] < 'A' || payment_new.clast[0] >
'Z') {
iprint (IPRINT_ERROR,
"ASSERTION: payment_new getname() returns
invalid name! '%s'\n",
payment_new.clast);
return RTE_ERROR;
}
payment_new.byname = 1;
payment_new.cid = 0;
} else {
payment_new.cid = NURand(1023, 1, 3000, CUSTC);
/* cust. # 1 to 3000 */
ordstat_new.cid = 0;
payment_new.clast[0] = (char) NULL;
}
whses = shmglobal->max_warehouses;
if (whses < 2 || uniform(1, 100) <= 85) /* for 85 % of
transactions */
ordstat_new.cid);
payment_new.cwid = WHSEID;
payment_new.cdoid = payment_new.did;
payment_new.remote = 0;
} else { /* for 15 % of transactions */
payment_new.cwid = (long) uniform((long)low_whse,
(long) whses-1);
if (payment_new.cwid >= WHSEID)
payment_new.cwid++;
payment_new.remote = 1;
payment_new.cdoid = uniform(1, 10); /* district 1 to 10
*/
}
dollars = uniform(1, 5000);/* dollar amt = 1 to 5000 */
if (dollars == 5000)
cents = 0;
else
cents = uniform(0, 99);
payment_new.amount = ((double) dollars) + ((double) cents) /
100.0;

//create payment results url
if (payment_new.byname)

sprintf(pymt_url,PAYMENT_RESULTS_CLAST_URL,term
inal,

payment_new.did,payment_new.clast,payment_new.cwid,

payment_new.cdoid,dollars,cents);
else

sprintf(pymt_url,PAYMENT_RESULTS_CID_URL,terminal
,

payment_new.did,payment_new.cid,payment_new.cwid,

```

```

payment_new.cdiddollarscents);
    sprintf(results_buffer,GET_REQUEST,pymt_url,host);

    // Go do the transaction
    rc = generic_transaction(&tran,host);
    payment = payment_new;
    payment.invalid = tran.invalid;
    //iprint(IPRINT_TRACE,"pymt txn finished, rc:%d
tran.invalid:%d\n",rc,tran.invalid);

    return (rc);
}
/*****
***          Stock Level Transaction          ***
*****/
int
StockLevel(char *host,int terminal)
{
    static struct stocklev_struct stocklevel, stocklevel_new;
    int rc;
    char *ptr;
    gen_tran_t tran;
    tran.invalid = 0;
    tran.data = &stocklevel;
    tran.len = sizeof(stocklevel);
    tran.keywait = 2;
    tran.type = STOCKLEV;
    tran.menu = "5";
    char stok_url[128];
    char form_buffer[256];
    char results_buffer[2048];
    tran.results_request = results_buffer;
    tran.form_request = form_buffer;
    //create stok form url
    sprintf(stok_url,STOCK_FORM_URL,terminal);
    sprintf(form_buffer,GET_REQUEST,stok_url,host);

    stocklevel_new.invalid = 0;
    stocklevel_new.threshold = uniform(10, 20); /* uniform no.
between 10 and          * 20 */

    //create stok results url
    sprintf(stok_url,STOCK_RESULTS_URL,terminal,stocklev
el_new.threshold);
    sprintf(results_buffer,GET_REQUEST,stok_url,host);

    // Go do the transaction
    rc = generic_transaction(&tran,host);
    stocklevel = stocklevel_new;
    stocklevel.invalid = tran.invalid;
    //iprint(IPRINT_TRACE,"stok txn finished, rc:%d
tran.invalid:%d\n",rc,tran.invalid);
    return (rc);
}
/*****
***          MAIN()          ***
*****/

```

```

int
user_transaction(char *host,void *data,int terminal)
{
    UserLocal *localdata = (UserLocal *)data;
    char logout[32];
    double ntask;
    int resp;
    static int task = 0;
    if (shmentry->flags & TES_FLAG_KEYSTROKE_TIME)
    {
        int rc;
        /* Wait for specified period of time */
        sleep (shmglobal->keystroke_sleep);
        /* Quit after one transaction */
        shm->lock(shmentry->pid);
        shmentry->flags |= TES_FLAG_DIE;

        shm->unlock(shmentry->pid);
        rc = NewOrder(host,terminal);
        iprint (IPRINT_INFO, "Slave %d: Keystroke timing
setting die flag\n", shmentry->num);
        return rc;
    }

    #if 1
    switch (shmglobal->test_state)
    {
        case 0: // Normal
            break;

        case 1: // pause
            sleep (1);
            return RTE_OK;

        case 2: // warmup
            switch(task++)
            {
                case 0: return Delivery(host,terminal);
                case 1: return OrderStatus(host,terminal);
                case 2: return Payment(host,terminal);
                case 3: return StockLevel(host,terminal);
                case 4: task = 0; return
NewOrder(host,terminal);
                return NewOrder(host,terminal);
            }
    }

    /*****
    ***          CHOOSE ONE OF THE TRANSACTIONS          ***
    *****/

    /*****
    ***
    ntask = (double) uniform(0.0, 100.0);
    if (ntask <= shmglobal->chances[DELIVERY])
    {
        return Delivery(host,terminal);
        //return NewOrder(host,terminal);
    }
    ntask -= shmglobal->chances[DELIVERY];
    if (ntask <= shmglobal->chances[ORDSTAT])
    {

```

```

        return OrderStatus(host,terminal);
    }
    ntask -= shmglobal->chances[ORDSTAT];
    if (ntask <= shmglobal->chances[PAYMENT])
    {
        return Payment(host,terminal);
    }
    ntask -= shmglobal->chances[PAYMENT];
    if (ntask <= shmglobal->chances[STOCKLEV])
    {
        return StockLevel(host,terminal);
    }
    return NewOrder(host,terminal);
}
#else
{
    int deck[100], count=-1, i, size=1, tmp;
    // lock deck
    if (count < 0) {
        // deck is empty fill it up
        count = 0;
        for (i = 0; i < 43 * size; i++) {
            deck[count++] = Payment;
        }
        for (i = 0; i < 4 * size; i++) {
            deck[count++] = StockLevel;
        }
        for (i = 0; i < 4 * size; i++) {
            deck[count++] = OrderStatus;
        }
        for (i = 0; i < 4 * size; i++) {
            deck[count++] = Delivery;
        }
        for (; count < 100 * size; i++) {
            deck[count++] = NewOrder;
        }
        // randomize the deck
        for (i = 0; i < 100 * size; i++) {
            int tmp;
            int pick = uniform(i+1, 100);
            tmp = deck[i];
            deck[i] = deck[pick];
            deck[pick] = tmp;
        }
    }
    tmp = deck[count-];
    // unlock deck
    switch(tmp) {
        case Delivery: return Delivery(host,terminal);
        case OrderStatus: return OrderStatus(host,terminal);
        case Payment: return Payment(host,terminal);
        case StockLevel: return StockLevel(host,terminal);
        case NewOrder: return NewOrder(host,terminal);
    }
    /*
    switch(tmp) {
        case Delivery: return Payment(host,terminal);
        case OrderStatus: return Payment(host,terminal);
        case Payment: return Payment(host,terminal);
        case StockLevel: return Payment(host,terminal);
        case NewOrder: return NewOrder(host,terminal);
    }
    */
}

```

```

}
#endif
#if 0
if (resp != RTE_OK) { /* logoff if response is not
correct */
strcpy(logout, "9\n"); /* menu option 9 */
transmit(logout);
resp = expect("tpcc_cstux_inf.");
return (ERROR);
} else
return (RTE_OK);
#endif
} /* end of Main */
int user_parameter_change(void) {
#if 0
int i;
iprint(IPRINT_TRACE, "Slave %d: total_users = %d\n",
shmentry->num);
iprint(IPRINT_TRACE, "Slave %d: chances = ", shmentry-
>num);
for (i = 0; i < MAX_TRAN_TYPE; i++)
iprint(IPRINT_TRACE, "%6.2f ", shmglobal->chances[i]);
iprint(IPRINT_TRACE, "\nSlave %d: think = ", shmentry-
>num);
for (i = 0; i < MAX_TRAN_TYPE; i++)
iprint(IPRINT_TRACE, "%6.2f ", shmglobal->think[i]);
#endif
return RTE_OK;
}
int user_login(char *user, char *password, void *data) {
UserLocal *localdata = (UserLocal *)data;
int rc;
int timeout_value = shmglobal->login_timeout;
char buffer[32];
set_typing_delay(0);
rc = expect (TRIGGER, timeout_value);
if (rc == RTE_ERROR) {
iprint (IPRINT_ERROR, "Slave %d: didn't find Warehouse
prompt\n", shmentry->num);
}
sprintf(buffer, "%d\t%d\n", localdata->Warehouse, localdata-
>District);
transmit(buffer);
iprint (IPRINT_TRACE, "Slave %d: Warehouse=%d,
District=%d, pid=%d\n", shmentry->num, localdata-
>Warehouse, localdata->District, getpid());
rc = expect (TRIGGER, timeout_value);
if (rc != RTE_OK) {
iprint (IPRINT_ERROR, "Slave %d: Failed logging in\n",
shmentry->num);
return RTE_ERROR;
}
return RTE_OK;
}
int user_login_html(char *host, void *data, int *terminal)
{
UserLocal *localdata = (UserLocal *)data;
int rc;
int timeout_value = shmglobal->login_timeout;
char request[256];

```

```

char url[30];
set_typing_delay(0);
iprint(IPRINT_ERROR, "Generating login request for
host:%s\n", host);
//generate login page request
sprintf(request, GET_REQUEST, LOGIN_URL, host);
iprint(IPRINT_ERROR, "sending login form
request:%s\n", request);
//send the request
transmit(request);
iprint(IPRINT_ERROR, "login request sent, reading response in
expect_html()\n");
//read the request
rc =
expect_html(LOGIN_TITLE, timeout_value, LOGIN_TITLE_LEN);
if (rc != RTE_OK)
{
iprint(IPRINT_ERROR, "Login request failed, unable
to find login key words:%s\n", LOGIN_TITLE);
return RTE_ERROR;
}
iprint(IPRINT_ERROR, "login request read\n");
//generate url and page get request
sprintf(url, MENU_URL, localdata->Warehouse, localdata-
>District);
sprintf(request, GET_REQUEST, url, host);
iprint(IPRINT_ERROR, "sending login results
request:%s\n", request);
transmit(request);
iprint (IPRINT_TRACE, "Slave %d: Warehouse=%d,
District=%d, pid=%d\n", shmentry->num, localdata-
>Warehouse, localdata->District, getpid());
rc = expect_html(MENU_TITLE,
timeout_value, MENU_TITLE_LEN);
if (rc != RTE_OK)
{
iprint (IPRINT_ERROR, "Slave %d: Failed logging in\n",
shmentry->num);
return RTE_ERROR;
}
iprint (IPRINT_TRACE, "User login successful Slave%d:
Warehouse=%d, District=%d, pid=%d\n", shmentry->num,
localdata->Warehouse, localdata->District, getpid());
rc = get_term_id(terminal);
if (rc != RTE_OK)
return RTE_ERROR;
iprint(IPRINT_TRACE, "Terminal set for this user:%d w/
warehouse:%d district:%d\n", *terminal, localdata-
>Warehouse, localdata->District);
return RTE_OK;
}
int get_term_id(int *terminal)
{
//search for terminal id
const char *termID_ptr;
if (!(termID_ptr = expect_buffer_search("NAME=\"01\"
VALUE=\"", 17)))
{

```

```

echo_trace ("Did not find terminal id in
response...");
iprint (IPRINT_ERROR, "No terminal id specified.");
return RTE_ERROR;
}
else
{
*terminal = atoi(termID_ptr+17);
iprint(IPRINT_ERROR, "Terminal id:%d\n", terminal);
}
return RTE_OK;
}
int user_init () {
extern int expect_save_active;
WHSEID = shmlocal->Warehouse;
status->max_transmit = shmglobal->keystroke_packet_size;
expect_save_active = 1;
return RTE_OK;
}
int user_logout () {
iprint (IPRINT_TRACE, "Slave %d: Warehouse=%d,
District=%d logging out\n", shmentry->num, shmlocal-
>Warehouse, shmlocal->District);
return RTE_OK;
}
int user_cleanup () {
transaction_sleep_do();
transaction_start(0, 0, NULL); // Just something to clear out
the buffer...
return RTE_OK;
}
int user_spawn_ok() {
int rc, hb;
hb = ((UserGlobal *) (shm->global_data))->host_busy;
rc = hb?RTE_ERROR:RTE_OK;
return rc;
}

```

user tpcc.h

```

/*****
*****
*/
/* user_tpcc.h Audit: 05/30/96 */
/*****
*****
*/
/* $Id: user_tpcc.h,v 1.1 1999/02/22 06:31:06 channui Exp $ */
#ifndef USER_TPCC_H
#define USER_TPCC_H
/*****
*****
*/
/*** run-time constant for customer last name from 0 to 255,
***/
/*** run-time constant for customer id from 0 to 1023, ***/
/*** run-time constant for item id from 0 to 8191. ***/
/*****
*****
*/
#define LASTC 117 */
/* Change for 3.1 */
#define LASTC 193

```

```

#define CUSTC 319
#define ITEMC 3849
/*****
*****/
/** response type */
/*****
*****/
/* #define OK 1 */
/* #define ERROR-1 */
/*****
*****/
/** transaction type */
/*****
*****/
#define NEWORDER 1
#define PAYMENT 2
#define ORDSTAT 3
#define DELIVERY 4
#define STOCKLEV 5
/*****
*****/
/** transaction structures */
/*****
*****/
struct neword_struct {
char invalid; /* transaction completed successfully */
long did;
long cid;
long oid; /* Order-ID returned from client */
long nloop; /* number of order line, avg = 15 */
char oremote; /* 1 for remote order, 10% */
long olremote; /* number of remote order line, 1% */
char rollback; /* actually saw rollback text on screen */
struct items_struct {
long olswid;
long oliid;
long olquantity;
} item[15];
};
struct payment_struct {
char invalid; /* transaction completed successfully */
long did;
long cid;
long cwid;
long cdid;
char clast[17];
double amount;
char byname; /* 1 for by last name, 0 for by id */
char remote; /* 1 for remote warehouse, 0
otherwise */
};
struct ordstat_struct {
char invalid; /* transaction completed successfully */
long did;
long cid;
char clast[17];
char byname; /* 1 for by last name, 0 for by id */
};
struct delivery_struct {
char invalid; /* transaction completed successfully */
char carrier;
};

```

```

struct stocklev_struct {
char invalid; /* transaction completed successfully */
long threshold;
};
struct generic_struct {
char invalid; /* transaction completed successfully */
};
union transaction_info {
char invalid;
struct generic_struct generic;
struct neword_struct neword;
struct payment_struct payment;
struct ordstat_struct ordstat;
struct delivery_struct delivery;
struct stocklev_struct stocklev;
};
struct UserGlobal {
int total_users;
int max_warehouses;
int keystroke_sleep;
int login_timeout;
int keystroke_packet_size;
int lastc;
int test_state;
int host_busy;
double chances[MAX_TRAN_TYPE];
double think[MAX_TRAN_TYPE];
double emulex_response[MAX_TRAN_TYPE];
double emulex_menu [MAX_TRAN_TYPE];
};
struct UserLocal {
int Warehouse;
int District;
};
struct user_data_header {
};
extern UserGlobal *shmglobal;
extern UserLocal *shmlocal;
#endif

```

Appendix - E: Third Party Pricing Information



800.750.4239

SHOPPING CART

[▶ Your Saved Carts](#) [▶ Save This Cart](#) [▶ Edit Saved Carts](#) [▶ Send To An Associate](#)

[Continue to Checkout](#)

Quantity	Product	CDW	Usually Ships	Price	Ext. Price
<input type="text" value="6"/>	NETGEAR GS116 16-port 10/100/1000Mbps Gigabit Switch	638864	Same Day	\$199.99	\$1,199.94

Click to remove an item from your cart

Sub-Total \$1,199.94

[Update](#)

[Clear Cart](#)

[Continue to Checkout](#)

[Continue Shopping](#) | [Go to CDW.com Homepage](#)

Related Top Sellers For: **NETGEAR GS116 16-port 10/100/1000Mbps Gigabit Switch**

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 425 882 8080
Fax 425 936 7329
<http://www.microsoft.com/>

Microsoft

January 22, 2006

IBM Corporation
Tony Petrossian
11501 BURNET ROAD
Austin, TX 78758

Mr. Petrossian:

Here is the information you requested regarding pricing for several Microsoft products to be used in conjunction with your TPC-C benchmark testing.

All pricing shown is in US Dollars (\$).

Part Number	Description	Unit Price	Quantity	Price
C11-00821	Windows 2000 Server <i>Server License Only - No CALs</i> <i>Discount Schedule: No Level</i> <i>Unit Price reflects a 8% discount from the retail unit price of \$799.</i>	\$738	41	\$30,258
254-00170	Visual C++ Standard Edition <i>No Discounts Applied</i>	\$109	1	\$109
N/A	Microsoft Problem Resolution Services <i>Professional Support</i> <i>(1 Incident)</i>	\$245	1	\$245

All products are currently orderable through Microsoft's normal distribution channels.

Some products may not be currently orderable but will be available through Microsoft's normal distribution channels by November 7, 2005.

Defect support is included in the purchase price. Additional support is available from Microsoft PSS on an incident by incident basis at \$245 per call.

This quote is valid for the next 90 days.

If we can be of any further assistance, please contact Jamie Reding at (425) 703-0510 or jamiere@microsoft.com.

Reference ID: PCtope0622016403.

Please include this Reference ID in any correspondence regarding this price quote.



International Business Machines Corporation

11400 BURNET RD
AUSTIN TX 78758

February 1, 2006

Dear Tony,

The requested quote for the p5-570 TPC-C benchmark is below.

Product	Description	Qty	List Price	Extended Purchase	Monthly Maint.	Extended Warr/Mnt
Server						
9117-570	Server 1:9117 Model 570	1	3,867	3,867	128	3,072
6942-25B	6051 - 9117-570 24x7x4 WSU	1	4,074			4,074
1846	Op Panel	1	199	199		
1849	Processor Cable, Four-Drawer System	1	17,868	17,868		
1859	SP Flex Cable, Four-Drawer System	1	7,412	7,412		
2640	IDE Slimline DVD-ROM Drive	1	274	274		
3274	73.4 GB 10,000 RPM Ultra320 SCSI Disk Drive	6	599	3,594		
4498	32GB (4x8GB) DIMMS (533MHZ DELHI) DDR2	16	175,000	2,800,000		
5716	2 Gigabit Fibre Channel PCI-X Adapter	6	1,999	11,994		
6001	SPCN 2m Cable	2	33	66		
6671	Power Cord (9-foot), Drawer to IBM PDU, 250V/10A	8	19	152		
7164	IBM Rack-mount Drawer Rail Kit	4	222	888		
8338	2-WAY 2.2GHZ GS DCM PROC, 36MB L3, 8 DDR2	8	10,720	85,760	176	33,792
7865	CEC Backplane	4	1,588	6,352		
7866	I/O Backplane	4	5,426	21,704		
7867	Midplane	4	662	2,648		
7868	DASD Backplane	4	1,588	6,352		
7869	Media Backplane	1	185	185		
7870	Power Midplane	4	265	1,060		
7768	PROCESSOR POWER REGULATOR	12	675	8,100		
7878	Serial Port Riser Card	4	132	528		
7879	System Drawer Enclosure	4	463	1,852		
7888	AC Power Supply, 240V, 1400 W	8	1,059	8,472		
7618	MODEL 570 PERMANENT PROCESSOR ACTIVATION FEATURE (FOR #8338)	16	21,440	343,040	155	59,520
7997	Service Processor Card (FSP)	1	860	860		
7014-T42	Rack 2:Rack Model T42	1	3970	3,970	37	888
6069	Front door (Black) for High Perforation (2m racks)	1	550	550		
6098	Side Panel (Black)	2	150	300		
6654	PDU to Wall Powercord 14', 200-240V/24A, UTG0247, PT#12	3	240	720		
7188	Power Dist Unit-Side Mount, Universal UTG0247 Connector	2	1200	2,400		
7311-D11	I/O Drawer 1:7311 Model D11	4	4461	17,844	174	16,704
6942-25B	60427311-D11 24x7x4 WSU	4	905			3,620
3147	RIO-2 (Remote I/O-2) Cable, 3.5M	4	728	2,912		
5716	2 Gigabit Fibre Channel PCI-X Adapter	16	1999	31,984		
6006	SPCN 3m Cable	4	53	212		
6276	AC Power Supply, 288 W	8	375	3,000		
6438	Rio-2 Remote I/O Loop Adapter	4	900	3,600		

6459	Power Cord (12ft) 250V/10A, RA	8	5	40		
6582	Planar Board, 6 PCI-X Slots	4	500	2,000		
6583	Air Blower	4	100	400		
7311	Dual I/O Unit Enclosure	4	417	1,668		
7310-C04	HMC 1:7310-C04 Desktop Hardw.Mgmt.Console	1	1830	1,830	48	1,152
6942-25B	7310-C03 24x7x4 WSU	1	192			192
3631	IBM ThinkVision C170 17-inch Color Monitor	1	250	250		
6470	Power Cord (6-foot), To Wall (125V, 15A), Plug Type #4	2	14	28		
7801	Ethernet Cable, 6M, Hardware Management Console to System Unit	1	12	12		
8800	Quiet Touch Keyboard - USB, Business Black, US English, #103P	1	83	83		
8841	Mouse - Business Black with Keyboard Attachment Cable	1	62	62		
			Subtotal	3,407,092	Subtotal	123,014

Software

5692-A5L	System Software	1	NC	NC		
1004	CD-ROM Process Charge	1	50	50		
967	MEDIA 5765-G03 AIX 5L V5.3	1	NC	NC		
968	Expansion pack	1	NC	NC		
970	AIX 5L V5.3 Update CD	1	NC	NC		
975	Microcode Upd Files and Disc Tool CD	1	NC	NC		
1403	Preinstall 64-bit Kernel	1	NC	NC		
2924	English Language	1	NC	NC		
3410	CD-ROM	1	NC	NC		
3506	Suppress Single Pallet Delivery	1	NC	NC		
5005	Preinstall	1	NC	NC		
5924	English U/L SBCS Secondary Language	1	NC	NC		
5765-G03	AIX 5L V5.3	1	NC	NC		
0005	Per Processor F5 AIX 5L V5.3	16	1,225	19,600		
5773-SM3	Software Maintenance for AIX, 3 Year	1	NC	NC		
0466	F5 3 Yr SWMA for AIX per Processor Reg/Ren	16	1,958			31,328
0468	F5 3 yr Services 7x24 Support per Processor	16	496			7,936
5773-RS3	HMC Initial Software Support 3 Year	1	NC	NC		
569	Per Processor Software Support 3 Year	1	675			675
570	Per Processor 24x7 Software Support 3 Year	1	236			236
D5A1DLL	C for AIX user Lic+SW maint 12 MO	1	515	515		
E1A1FLL	C for AIX user annual SW maint renewal	2	103			206
	DB2 Enterprise Server Edition Proc Lic/1 yr Maint.	16	23,902	382,432		
E00BILL	DB2 Enterprise Server Ed Proc Maint Renew	32	1,138			36,416
			Subtotal	402,597	Subtotal	76,797

Storage

1740-710	DS4000 EXP710 Storage Expansion	124	6,000	744,000		
2210	(19K1271) 2Gb Fibre Channel Short Wave GBIC	518	499	258,482		
5212	(06P5772) 2Gb FC, 36.4GB/15K Drive	1,688	1,115	1,882,120		
5601	(19K1247) 1m 50u Fiber Optic Cable (LC-LC)	248	79	19,592		
6942-25B	74231740-710 24x7x4 WSU	124	760			94240
1815-82A	DS4800 Disk System Model 82 (4 GB Cache)	11	53,995	593,945		
7711	(22R4255) DS4800 AIX Host Kit	1	7,000	7,000		
8870	DS4800 8-Storage Partitions	11	10,000	110,000		
5605	(19K1248) Fiber Cable 5m Multimode (LC-LC)	22	129	2,838		
6942-25B	47371815-82A 24x7x4 WSU	11	3,200			35200
			Subtotal	3,617,977	Subtotal	129,440

Client

8837E3U	xSeries 336	40	3,319	132,760	
30R5096	36GB 15K U320 SCSI HS Express	40	269	10,760	
93074SX	NetBay42 Standard Rack	11	1,489	16,379	
90P0744	Optical 3-Button Mouse - USB	1	15	15	
25R6968	Preferred Pro Full Size PS/2 Keyboard	1	29	29	
49387NU	IBM C117 17" CRT Monitor	1	149	149	
21P2073	Warranty upgrade; 3 year onsite repair 24x7x4 hour	40	450		18000
			Subtotal	160,092	Subtotal
					18,000
			Total	7,587,758	347,251
					(3,412,054)
					4,522,955

For additional information, please contact me directly at 1-512-838-6804.

Charles J. Gitomer
 pSeries Offering Manager
 IBM Corporation
 gitomer@us.ibm.com