TPC Benchmark™ D
Full Disclosure Report

Data General Corporation

AViiON™ Model AV6600
Using Oracle 7.3.3
and Windows NT 4.0

First Edition
June 1997

TPC Benchmark™ D Full Disclosure Report
Data General AV6600 Server Using Oracle 7.3.3 and Windows NT 4.0
Edition 1

First Printing June 1997

Printed in U.S.A.

Data General Corporation (DGC) believes that the technical, pricing and discounting information in this document is accurate as of its publication date. The performance information in this document is for guidance only. System performance is highly dependent on many factors including system hardware, system and user software, and user-application characteristics. Customer applications must be carefully evaluated before estimating performance. DGC does not warrant or represent that a user can or will achieve similar performance as expressed in this document.

THE TERMS AND CONDITIONS GOVERNING THE SALE OF DGC HARDWARE PRODUCTS AND THE LICENSING OF DGC SOFTWARE CONSIST SOLELY OF THOSE SET FORTH IN THE WRITTEN CONTRACTS BETWEEN DGC AND ITS CUSTOMERS. NO REPRESENTATION OR OTHER AFFIRMATION OF FACT CONTAINED IN THIS DOCUMENT INCLUDING BUT NOT LIMITED TO STATEMENTS REGARDING PRICE, CAPACITY, RESPONSE-TIME PERFORMANCE, SUITABILITY FOR USE, OR PERFORMANCE OF PRODUCTS DESCRIBED HEREIN SHALL BE DEEMED TO BE A WARRANTY BY DGC FOR ANY PURPOSE, OR GIVE RISE TO ANY LIABILITY OF DGC WHATSOEVER.

DGC assumes no responsibility for any errors that may appear in this document. DGC reserves the right to make changes in specifications and other information contained in this document without prior notice, and the reader should in all cases consult DGC to determine whether any such changes have been made.

AViiON, and CLARiiON are U.S. registered trademarks of the Data General Corporation.

ORACLE, SQL*Loader, Oracle7, and Pro*C are registered trademarks of Oracle Corporation.

Windws NT is a trademark of the Microsoft Corporation.

TPC Benchmark is a trademark of the Transaction Processing Performance Council.

Additional Copies

Please use the toll-free number below to request additional copies of this report.

Telephone: 1-800-DATAGEN

Document Title: TPC Benchmark™ D Full Disclosure Report
Data General AV6600 Server Using Oracle 7.3.2.3 and Windows NT 4.0

Edition 1

Abstract

This report documents the methodology and results of the TPC Benchmark D test conducted on the AV6600 using Oracle 7.3.3, in conformance with the requirements of the TPC-D Benchmark Specification. The operating system used for the benchmark was Windows NT 4.0 Service Pack 2. The application was written in C and compiled using Microsoft Visual C++ for NT.

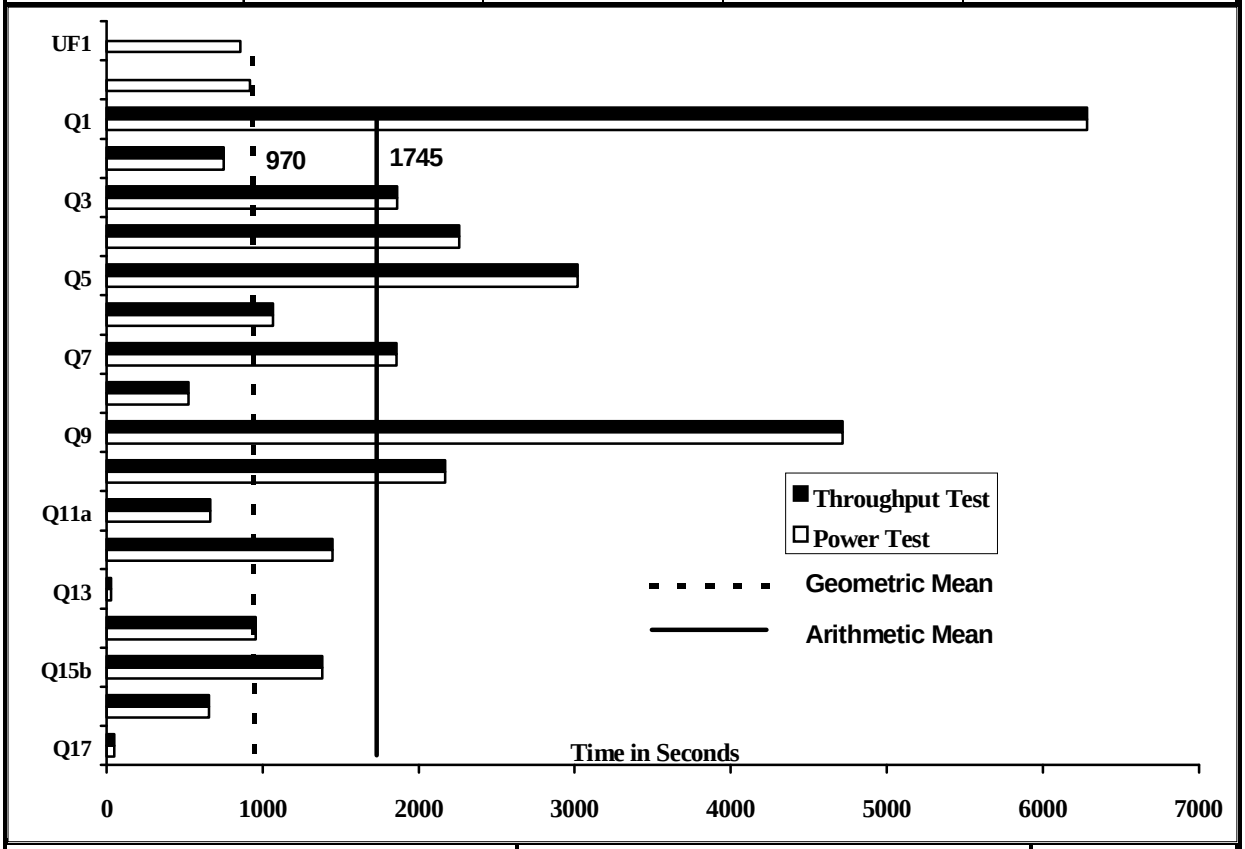
The TPC-D Benchmark was developed by the Transaction Processing Performance Council (TPC). The TPC was founded to define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry.

The benchmark configuration, environment and methodology used to produce and validate the test results, and the pricing model used to calculate the price/performance were audited by Francois Raab, Information Paradigm, Inc., to verify compliance with the relevant TPC specifications. The auditor's letter of attestation is attached as Appendix G.

Data General Corporation	AViiON Model AV6600	TPC-D REV 1.2.2 Report Date: 11-Jun-97

TOTAL SYSTEM COST	TPC-D POWER	TPC-D THROUGHPUT	PRICE/PERFORMANCE
581,046	370.9 QppD @ 100GB	194.6 QthD @ 100GB	2,163 $ per QphD @ 100GB

DATABASE SIZE	DATABASE MANAGER	OPERATING SYSTEM	OTHER SOFTWARE	AVAILABILITY DATE
100GB	Oracle Version 7.3.3	Microsoft Windows NT 4.0 SP2	MKS Toolkit NT Resource Kit Visual C++	1-Jul-97



DATABASE LOAD TIME = 15:50:50	DISK SIZE/DATABASE SIZE = 3.59	RAID: Y

SYSTEM COMPONENTS	DESCRIPTION
NUMBER OF NODES	Data General AViiON AV6600
PROCESSORS	6 x 200MHz PentiumPro, 512K Cache
MEMORY	2 GB
DISK DRIVES	83 x 4.3 GB and 1 x 2.1 GB
TOTAL GB OF STORAGE	359 GB

Data General Corporation	AViiON Model AV6600	TPC D REV 1.2.2 Report Date: 11-Jun-97

Description	Part Number	Unit Price	Qty	Extended Price	5 yr. Maint. Price
Server Hardware					
AV6600,3 CPU, 256MB, LAN, SCSI, CD	70706-AE	30,547	1	30,547	9,196
ADD-ON 512MB MEM (4X128MB DIMM)	7105	15,360	3	46,080	4,860
REPL 256MB WITH 512MB (4X128MB DIMM)	R7374	10,816	1	10,816	1,426
ADD-ON TRI-CPU BOARD (200MHZ/512K)	7100	12,997	1	12,997	1,497
D1600I 14" TERMINAL, WHITE, ERGONOMIC	6945W	293	1	293	357
D1200I/D1600I 101-KEYBOARD, PWR CORD	G6001A-A	56	1	56	107
60" DEEPRACK CABINET	14001-G7	1,772	1	1,772	0
2 channel SCSI controller	3944-AUWD	715	4	2,860	2,048
			Subtotal:	**105,421**	**19,491**
Server Software					
ORACLE FOR NT	ORACLE NT	43,056	1	43,056	73,440
WIN NT SVR 4,1-8 CPU PRE-INSTALL LMD	Q138AG21CD	2,400	1	2,400	10,560
Windows NT Resource Kit	NT RESRC KIT	499	1	499	0
MKS Toolkit for NT	MKS NT	499	1	499	0
Microsoft Visual C++ v. 4.0	VISUAL C++ 4.0	499	1	499	2,852
			Subtotal:	**46,953**	**86,852**
Storage Devices					
4GB 1" HOT SWAP DISK FOR AV3600R/6600	61024-SJC	1,102	1	1,102	571
2GB 1" HOT SWAP DISK FOR AV3600R/6600	61024-SJC	777	1	777	571
HIGH PERFORMANCE ARRAY PACKAGE	79301R-E	28,574	4	114,296	26,876
NT ATTACHMENT KIT FOR CLARiiON	7945	488	1	488	0
REPLACE 8MB CACHE WITH 64MB	R7341	4,004	4	16,016	8,447
4GB 7200 RPM Disk	79012HC	1,313	82	107,666	45,519
			Subtotal:	**240,345**	**81,984**
			Total:	**392,719**	**188,327**
			Five Year Cost of Ownership:		**581,046**

Audited By: Francois Raab of Information Paradigm, Inc.

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 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.

Data General Corporation	AViiON Model AV6600	TPC-D REV 1.2.2
		Report Date: 11-Jun-97

Numerical Quantities

Measurement Results:

Database Scaling (SF/Size)	=	100
Total Data Storage / Database Size	=	3.59
Database Load Time	=	15 hours 50 minutes 50 seconds
Query Streams for Throughput Test	=	0
TPC-D Power Metric (QppD@100GB)	=	370.9
TPC-D Throughput Metric (QthD@100GB)	=	194.6
Total System Price Over 5 Years	=	$581,046
TPC-D Price Performance Metric ($/QphD@100GB)	=	$2,163

Measurement Intervals

Measurement interval in Throughput Test (TS)	=	31443 seconds

Duration of Stream Execution:

Stream ID	Seed	Start Date	Start Time	End Date	End Time	Total Time
UF1		6/2/97	1:16:14	6/2/97	1:30:27	0:14:13
Stream 00	1123198960	6/2/97	1:16:14	6/2/97	10:00:17	9:44:03
UF2		6/2/97	9:45:01	6/2/97	10:00:17	0:15:16

Timing Intervals (in Seconds)

Query	1	2	3	4	5	6	7	8	9
Stream 00	6285.4	750.5	1861.5	2256.9	3017.7	1066.0	1854.2	522.0	4715.0

Query	10	11	12	13	14	15	16	17	UF1	UF2
Stream 00	2167.4	661.8	1446.1	28.8	954.0	1381.7	654.6	48.3	853.2	916.3

Measurement Intervals

Measurement interval in throughput test	14,066 seconds

Duration of stream execution

Stream ID	Seed	Start Date	Start Time	End Date	End Time	Total Time
Stream 00	458677759	5/15/97	16:28:48	5/15/97	20:53:51	4:25:03
UF1		5/15/97	16:28:48	5/15/97	16:43:02	
UF2		5/15/97	20:37:29	5/15/97	20:53:51	

Timing Intervals (in seconds)

q1	q2a	q3	q4b	q5	q6	q7	q8b	q9	q10
2829.5	313.9	841.9	1033.5	1535.8	508.7	841.4	300.2	2207.7	1079.3

q11	q12b	q13	q14c	q15b	q16	q17	uf1	uf2
273.4	727.8	28.0	544.2	664.2	296.2	40.4	854.2	982.4

Table of Contents

General Items

Test Sponsor

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

Data General Corporation is the sponsor of this TPC Benchmark™ D.

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:

- *Database Tuning Options*
- *Optimizer/Query execution options*
- *Query processing tool/language configuration parameters*
- *Recovery/commit options*
- *Consistency/locking options*
- *Operating system and configuration parameter*
- *Configuration parameters and options for any other software in the pricing structure*
- *Compiler optimization options*

This requirement can be satisfied by providing a full list of all parameters and options, as long as all those which have been modified from their default values have been clearly identified and these parameters and options are only set once.

Details of system and database configurations and parameters are provided in Appendix A.

Configuration Items

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

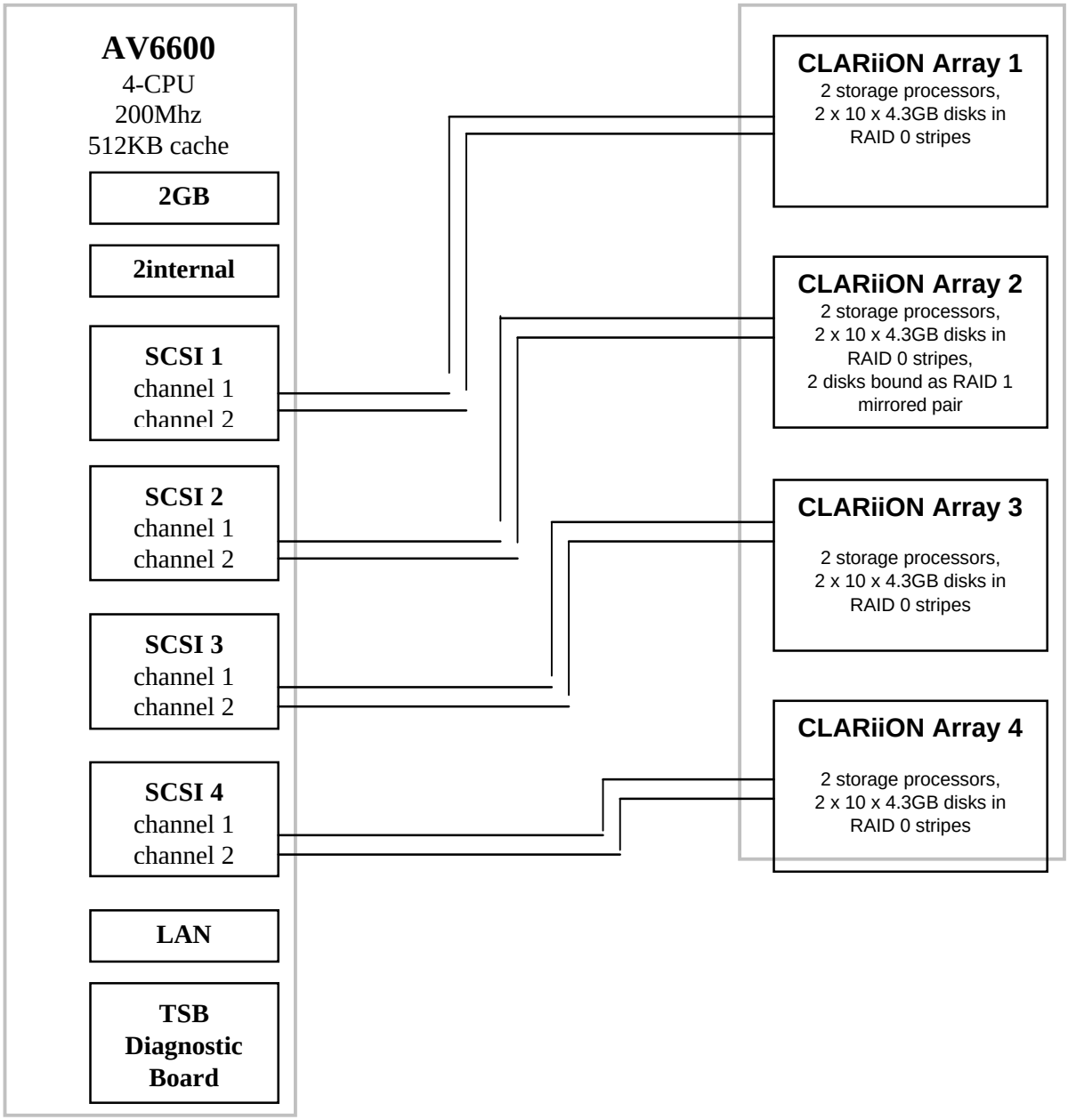
- *Number and type of processors*
- *Size of allocated memory, and any specific mapping/partitioning of memory in the test.*
- *Number and type of disk units (and controllers, if applicable).*
- *Number of channels or bus connections to disk units, including their protocol type.*
- *Number of LAN (e.g. Ethernet) Connections, including routers, workstations, terminals, etc., that were physically used in the test or are incorporated into the pricing structure.*
- *Type and the run-time execution location of software components (e.g., DBMS, query processing tools/languages, middle-ware components, software drivers, etc.).*

The server System Under Test (SUT), a Data General AViiON AV6600, depicted in the next diagram, consists of:

- (6) 200MHz PentiumPro® Processors, each with 512KB of L2 cache.

- 2GB of main memory.
- 4 SCSI dual controllers (plus an internal SCSI adapter).
- 1 Ethernet controller.
- 2 internal disk drives.
- 4 CLARiiON Disk Arrays, each with two storage processors; 3 arrays with 20 disk drives and one with 22 disk drives.

The system's disk I/O configuration consists of 5 SCSI buses. The internal SCSI adapter controls the 2 internal disks. Each of the other 8 SCSI channels (2 on each adapter) are connected to 1 Storage Processor controlling a 10 disk RAID 0 stripe; except for 1 channel, which in addition controls a pair of mirrored RAID 1 disks.



Modem*

Measured and Priced Configuration

***** The only difference betwwen the measured and priced systems is that the measured system did not have a modem installed.

Clause 1: Logical Database Design

Table Definitions

Listings must be provided for all table definition statements and all other statements used to set up the test and qualification databases.

Appendix B contains the scripts that create and analyze the tables and indexes for the TPC-D database.

Physical Organization of Database

The physical organization of tables and indices, within the test and qualification databases, must be disclosed. If the column ordering of any table is different from that specified in Clause 1.4, it must be noted.

No record clustering or index clustering was used for this benchmark.

Horizontal Partitioning

Horizontal partitioning of tables and rows in the test and qualification databases (see Clause 1.5.4) must be disclosed.

The database layout is described in the tables in Clause 4.

Replication

Any replication of physical objects must be disclosed and must conform to the requirements of Clause 1.5.6.

Replication was not used for this benchmark.

Clause 2: Queries and Update Functions

Query Language

The query language used to implement the queries must be identified.

SQL was the query language used to implement all queries.

Random Number Generation

The method of verification for the random number generation must be described unless the supplied DBGEN and QGEN were used.

The 1.1.0A version of DBGEN and version 1.2.0 of QGEN were used to generate the random numbers for this TPC-D benchmark.

Substitution Parameters Generation

The method used to generate values for substitution parameters must be disclosed. If QGEN is not used for this purpose, then the source code of any non-commercial tool used must be disclosed. If QGEN is used, the version number, release number, modification number and patch level of QGEN must be disclosed.

The 1.1.0B version of QGEN was used to generate the substitution parameters.

Query Text and Output Data from Database

The executable query text used for query validation must be disclosed along with the corresponding output data generated during the execution of the query text against the qualification database. If minor modifications (see Clause 2.2.3) have been applied to any functional query definitions or approved variants in order to obtain executable query text, these modifications must be disclosed and justified. The justification for a particular minor query modification can apply collectively to all queries for which it has been used. The output data for the power and throughput tests must be made available electronically upon request.

Appendix C contains the query text and query output. The minor query modifications used in this implementation include the following:

- Wherever there is a date given in the WHERE clause of a select the TO_DATE function is used to convert the character input to the proper date format.
- Wherever there is a date returned by the query the TO_CHAR function is used to convert the date format to the proper output format.
- The Order table defined in the specification has been named the orders table due to conflict with a reserved word.
- Whenever there is a time modification (i.e. add 3 months to a date) the ADD_MONTHS function is used to calculate and adjust the time delta and keep it in the date/time format.

- CREATE TABLE statements have specific storage parameters in them. Namely, what tablespace to create them on, the sizes of the extents, and how full the extents should be before allocating a new one. The degree of parallelism was not specified for any temporary tables created.

Query Substitution Parameters and Seeds Used

All the query substitution parameters used during the performance test must be disclosed in tabular format, along with the seeds used to generate these parameters.

Appendix D contains the seed and query substitution parameters.

Update Function Source Code

The details of how the update functions were implemented must be disclosed (including source code of any non-commercial program used).

The update function is part of the implementation-specific driver code included in Appendix E.

Database Maintenance Option

The details of the database maintenance option selected (i.e., reset or evolve) must be disclosed (including source code of any non-commercial program used).

This implementation of the TPC-D benchmark uses the reset option.

Clause 3: Database System Properties

Atomicity

The results of the ACID tests must be disclosed along with a description of how the ACID requirements were met. This includes disclosing the code written to implement the ACID Transaction and Query.

Completed Transaction

Perform the ACID Transaction for a randomly selected set of input data and verify that the appropriate rows have been changed in the ORDER, LINEITEM, and HISTORY tables.

1. The total prices from the ORDER table and the extended price from the LINEITEM table were retrieved for a randomly selected order key.
2. The ACID Transaction was performed using the order key from step 1.
3. The ACID Transaction was committed.
4. The total price from the ORDER table and the extended price from the LINEITEM table were retrieved for the same order key. It was verified that the appropriate rows had been changed.

Aborted Transaction

Perform the ACID transaction for a randomly selected set of input data, substituting a ROLLBACK of the transaction for the COMMIT of the transaction. Verify that the appropriate rows have not been changed in the ORDER, LINEITEM, and HISTORY tables.

1. The total price from the ORDER table and the extended price from the LINEITEM table were retrieved for a randomly selected order key.
2. The ACID Transaction was performed using the order key from step 1. The transaction was stopped prior to the commit.
3. The ACID Transaction was rolled back.
4. The total price from the ORDER table and the extended price from the LINEITEM table were retrieved for the same order key, and were verified to have not been changed.

Consistency

Consistency is the property of the application that requires any execution of transactions to take the database from one consistent state to another.

Consistency Test

Verify that ORDER and LINEITEM tables are initially consistent, submit the prescribed number of ACID Transactions with randomly selected input parameters, and re-verify the consistency of the ORDER and LINEITEM tables.

1. The consistency of the ORDER and LINEITEM tables was verified based on a sample of O_ORDERKEYs.

2. 100 ACID transactions were submitted from each of 2 execution streams.
3. The consistency of the ORDER and LINEITEM tables was verified a second time with the same O_ORDERKEYs.

Isolation

Operations of concurrent 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.

Read-Write Conflict with Commit

Demonstrate isolation for the read-write conflict of a read-write transaction and a read-only transaction when the read-write transaction is committed.

1. An ACID transaction was started for a randomly selected O_KEY, L_KEY, and DELTA. The ACID transaction as suspended prior to commit.
2. An ACID query was started for the same O_KEY used in step 1. The ACID query completed and did not see the uncommitted changes made by the ACID transaction.
3. The ACID transaction was committed.

Read-Write Conflict with Rollback

Demonstrate isolation for the read-write conflict of a read-write transaction and a read-only transaction when the read-write transaction is rolled back.

1. An ACID transaction was started for a randomly selected O_KEY, L_KEY, and DELTA. The ACID transaction was suspended prior to rollback.
2. An ACID query was started for the same O_KEY used in step 1. The ACID query completed and did not see the uncommitted changes made by the ACID transaction.
3. The ACID transaction was rolled back.

Write-Write Conflict with Commit

Demonstrate isolation for the write-write conflict of two update transactions when the first transaction is committed.

1. An ACID transaction, T1, was started for a randomly selected O_KEY, L_KEY, and DELTA. The ACID transaction was suspended prior to commit.
2. A second ACID transaction, T2, was started using the same O_KEY and L_KEY and a different randomly selected DELTA.
3. T2 waited.
4. T1 was allowed to commit and then T2 completed.
5. It was verified that T2.L_EXTENDPRICE was calculated correctly.

Write-Write Conflict with Rollback

Demonstrate isolation for the write-write conflict of two update transactions when the first transaction is rolled back.

1. An ACID transaction, T1, was started for a randomly selected O_KEY, L_KEY, and DELTA. The ACID transaction was suspended prior to rollback.
2. A second ACID transaction, T2, was started using the same O_KEY and L_KEY and a different randomly selected DELTA.
3. T2 waited.
4. T1 was allowed to rollback and then T2 completed.
5. It was verified that T2.L_EXTENDPRICE was calculated correctly.

Durability

The tested system must guarantee durability: the ability to preserve the effects of committed transactions and insure database consistency after recovery from any one of the failures listed in Clause 3.5.2.

Failure of a Durable Medium

Guarantee the database and committed updates are preserved across a permanent irrecoverable failure of any single durable medium containing TPC-D database tables or recovery log tables.

The database logs were stored on a hardware mirrored pair (2 disks running RAID 1).

The tables for the database were stored on RAID-0 stripes.

1. The datafiles were backed up to an alternate disk media.
2. Two streams of ACID transactions were started.
3. While the test was running one side of the mirrored set of logs was disabled.
4. After it was determined that the test would still run with the loss of a log disk, a data disk was disabled.
5. The two streams of ACID transactions failed and recorded their numbers of committed transactions in success files.
6. The database was brought down.
7. The datafiles were restored to their state prior to the ACID transaction streams.
8. The database ran through its recovery mode.
9. The counts in the success files and the HISTORY table count were compared. The counts matched.

System Crash

Guarantee the database and committed updates are preserved across an instantaneous interruption (system crash/system hang) in processing which requires the system to reboot to recover.

1. Two streams of ACID transactions were started.
2. While the streams of ACID transactions were running the system was powered off.
3. When power was restored the system rebooted and the database was restarted.
4. The database went through a recovery period.
5. The success file and the HISTORY table counts were compared, and they matched.

Memory Failure

Guarantee the database and committed updates are preserved across failure of all or part of memory (loss of contents).

The system crash test and the memory failure test were combined. See the previous section.

Clause 4: Scaling and Database Population

Initial Cardinality of Tables

The cardinality (e.g., the number of rows) of each table of the test database, as it existed at the completion of the database load (see clause 4.2.5) must be disclosed.

Initial number of rows

Table	Occurrences
Orders	150,000,000
Lineitem	600,037,902
Customer	15,000,000
Part	20,000,000
Supplier	1,000,000
Partsupp	80,000,000
Nation	25
Region	5

Distribution of Tables and Logs Across Media

The distribution of tables and logs across all media must be explicitly described.

The database tables were distributed across 4 CLARiiON disk arrays. Each array contained two storage processors. The arrays were configured as follows:

- Database files were stored on 8 RAID 0 stripes, each consisting of (10) 4.3GB disks.
- Log information was stored on a mirrored pair, using (2) 4.3GB disks.
- Flat files were stored on (28) 4.3GB disks, which were physically removed before measurement.
- Database binaries were stored on one 4.3GB disk.
- The operating system was stored on one 2.1GB internal disk internal.

In the following tables, each row in the Controller column refers to one of two storage processors in each of the 4 CLARiiON disk arrays.

Distribution of Storage

Controller	Disk Drive	Storage Content
1	1-10	1 RAID 0 stripe for the database datafiles
2	11-20	1 RAID 0 stripe for the database datafiles

3	1-10	1 RAID 0 stripe for the database datafiles
4	11-20	1 RAID 0 stripe for the database datafiles
4	21,22	Mirrored pair containing one log
5	1-10	1 RAID 0 stripe for the database datafiles
6	11-20	1 RAID 0 stripe for the database datafiles
7	1-10	1 RAID 0 stripe for the database datafiles
8	1-7	1 RAID 0 stripe for the database datafiles

Partitioning and Replication

The mapping of database partitions/replications must be explicitly described. Implementations may use some form of RAID to ensure high availability. If used for data, auxiliary storage (e.g. indexes) or temporary space, the level of RAID used must be disclosed for each device.

No replication was used for this implementation.

Distribution of Partitioned Tables

Controller	Configuration	Usage	Fraction
1	RAID 0	lineitem, orders, parts, partsupp, supplier, nation, and region	1/8th of each of these tables except nation and region 100%
2	RAID 0	lineitem, orders, parts, partsupp, supplier	1/8th of each of these tables
3	RAID 0	lineitem, orders, parts, partsupp, supplier	1/8th of each of these tables
4	RAID 0	lineitem, orders, parts, partsupp, supplier	1/8th of each of these tables
4	Mirrored pair	Log	100%
5	RAID 0	lineitem, orders, parts, partsupp, supplier	1/8th of each of these tables
6	RAID 0	lineitem, orders, parts, partsupp, supplier	1/8th of each of these tables
7	RAID 0	lineitem, orders, parts, partsupp, supplier	1/8th of each of these tables
8	RAID 0	lineitem, orders, parts, partsupp, supplier	1/8th of each of these tables

DBGEN Version and Modifications

The version number, release number, modification number, and patch level of DBGEN must be disclosed. Any modifications to the DBGEN (see Clause 4.2.1) source code….must be disclosed. In the event that a program other than DBGEN was used to populate the database, it must be disclosed in its entirety.

The supplied DBGEN Version 1.1.0A was used for database population.

Database Content of the First Ten Rows

The content of the first ten rows of each table in the test database must be disclosed.

Appendix F contains the first ten rows of each table in the test database.

Database Load time

The database load time for the test database (see clause 4.3) must be disclosed.

Database load time was 15 hours 50 minutes 50 seconds.

Data Storage Ratio

The data storage ratio must be disclosed. It is computed by dividing the total data storage of the priced configuration (expressed in GB) by the size chosen for the test database as defined in 4.1.3.1. The ratio must be reported to the nearest 1/100, rounded up.

Data Storage Ratio

Disk Type	Number of Disks	Space per Disk	Subtotal Disk Space
4.3GB	83	4.3GB	356.9GB
2.1GB	1	2.1GB	2.1GB

Total disk storage: 359.0GB

Data storage ratio: 3.59

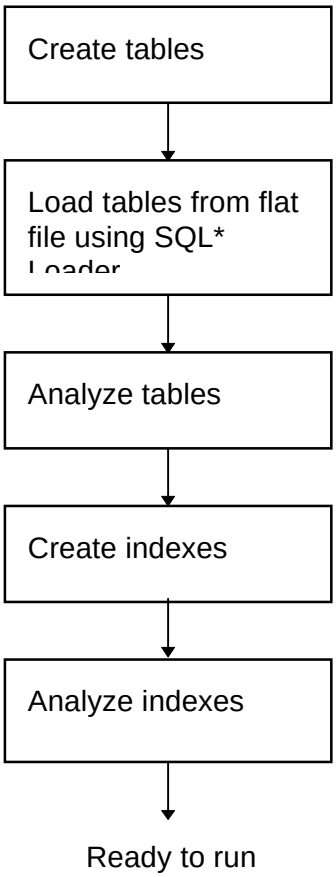
Database Load Mechanism Details and Illustration

The details of the database load must be disclosed, including a block diagram illustrating the overall process. Disclosure of the load procedure includes all steps, scripts, input and configuration files required to completely reproduce the test and qualification databases.

DBGEN was used to create flat files which were then loaded into the tablespaces using SQL*Loader. The flat files were split into 40 files per table, except the REGION and NATION tables which each had only one file.

PERL was used to perform the creation of the tablespaces, tables, and indexes of the database. It was also used to run the analyze scripts. The controller cache was set to 0MB write cache and 60 MB read cache during the database load and the power runs. This was done manually, through running of the CLARiiON Dass Manager.

After the load phase was completed, the flat file disks were physically removed from the system.



Database Load Process

Clause 5: Performance Metrics and Execution Rules

Steps in the Power Test

The details of the steps followed to implement the power test (e.g., system boot, database restart, etc.) must be disclosed.

The following steps were used to implement the power test:

1. Database restart.
2. UF1 update transaction.
3. Stream 00 execution.
4. UF2 update transaction.

Timing Intervals

The timing intervals (see Clause 5.3.6) for each query of the measured set and for both update functions must be reported for the power test.

The power test timing intervals are disclosed in the Numerical Quantities Summary earlier in this document.

Number of Streams for The Throughput Test

The number of execution streams used for the throughput test must be disclosed.

No separate throughput test was conducted. Therefore, the values of power test were used for the throughput test.

Start/Finish Time of Each Query Stream

The start time and finish time for each query execution stream must be reported for the throughput test.

The throughput test start time and finish time for each stream are disclosed in the Numerical Quantities Summary earlier in this document.

Total Elapsed Time

The total elapsed time of the measurement interval must be reported for the throughput test.

The total elapsed time of the throughput test was 31,443 seconds.

Start/Finish Time for Update Function

Start and finish time for each update function in the update stream must be reported for the throughput test.

The start and finish time for each update function in the update stream are disclosed in the Numerical Quantities Summary earlier in this document.

Timing Intervals for Each Query and Each Update

The timing intervals (see Clause 5.3.6) for each query of each stream and for each update function must be reported for the throughput test.

The timing intervals for each query and each update function are contained in the Numerical Quantities Summary disclosed earlier in this document. Since no separate throughput test was conducted, the values for the power test are used for the throughput test.

Performance Metrics

The computed performance metrics, related numerical quantities and the price performance metric must be reported.

The performance metrics, and the numbers on which they are based, are contained in the Numerical Quantities section of the Executive Summary.

Reproducibility Method

A description of the method used to determine the reproducibility of the measurement results must be reported. This must include the performance metrics (QppD and QthD) from the reproducibility runs.

Performance results from the first two executions of the TPC-D benchmark indicated the following percent differences for the metrics:

Percentage Differences in Benchmark Executions

Run	QppD@100GB	QthD@100GB	QphD@100GB
1	370.9	194.6	268.7
2	372.4	194.6	269.2
Percent Difference	0.40%	0.02%	0.19%

Clause 6: SUT and Driver Implementation

Driver

A detailed description of how the driver performs its functions must be supplied, including any related source code or scripts. This description should allow an independent reconstruction of the driver.

QGEN is first called with a stream ID of 0 to generate the QET for the power test. The power test is performed by a shell script called **runpower1.sh**. This shell script calls **runuf1.sh**, which performs the update function. Queries are then executed with the **qexec.ott** ISL program. The last part of the power test is run through another shell script called **runuf2.sh**. Both wall-clock and high-resolution times are collected for the measurement intervals.

Update functions are parallelized by partitioning the data set and utilizing multiple processes to perform the updates. Two Pro*C programs are written to perform the inserts and deletes required by UF1 and UF2. Batch array inserts are used for UF1 and logical consistency is ensured by limiting the batching of ORDER and LINEITEM rows to ones that have the same order key ranges.

Implementation Specific Layer (ISL)

If an implementation-specific layer is used, then a detailed description of how it performs its functions must be supplied, including any related source code or scripts. This description should allow an independent reconstruction of the implementation-specific layer.

Query execution text generated by QGEN is picked up by the ISL program, **qexec.ott**, which submits the query to the SUT. The ISL program utilize the Oracle Call Interface (OCI) to communicate with the Oracle database on the SUT. QETs directly generated by QGEN are read and submitted to the SUT via the ISL program as dynamic SQL statements. The ISL program then fetches the query execution output and reports it to the user. Timings are taken at intervals specified by Clause 5.3.6.2.

Clause 7: Pricing

Hardware and Software Used in the Priced System

A detailed list of 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) of price(s) must also be reported.

A detailed list of the hardware and software used in the priced system is included in the executive summary at the beginning of this document.

Total Five year Price

The total 5-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.

A detailed price sheet of all the hardware and software used in this configuration, including the 5-year maintenance cost, and total price, is included in the executive summary at the beginning of this document. All maintenance has been uplifted from 8x40 to 7x24. All AViiON hardware comes with a 1 year 48-hour response warrantee, which has been uplifted to 7x24 4-hour response. Software comes with a 90 day telephone warantee, which has been uplifted to 7x24 support with updates. Oracle is discounted 40%, and the maintenance 15% off the base prices.

Availability Date

The committed delivery date for general availability of products used in the price calculations must be reported. When the priced system includes products with different availability dates, the availability date reported on the executive summary must be the date by which all components are committed to being available. The full disclosure report must report availability dates individually for at least each of the categories for which a pricing subtotal must be provided.

All components are available as of the date reported in the executive summary.

Clause 8: Auditor-Related Items

Auditor's Report

The auditor's agency name, address, phone number, and Attestation letter with a brief audit summary report indicating compliance must be included in the full disclosure report. A statement should be included specifying who to contact in order to obtain further information regarding the audit process.

The auditor's letter of attestation is included in this full disclosure report as Appendix G.

This implementation was audited by: Francois Raab, Information Paradigm, TPC Certified D Auditor.

Further information regarding the audit process may be obtained from:

Information Paradigm Inc.
1373 North Franklin Street
Colorado Springs, CO 80903
719/473-7555

Requests for this TPC Benchmark D Full Disclosure Report should be sent to:
Transaction Processing Performance Council
c/o Shanley Public Relations
777 North First Street, Suite 6000
San Jose, CA 95112-6311
408/295-8894

Appendix A: Parameter Settings

System Configuration Settings

The following NT services were started:

- Alerter
- EventLog
- License Logging Service
- Messanger
- Net Logon
- OracleServiceTPCD
- Plug and Play
- Remote Procedure Call (RPC) Locator
- Remote Procedure Call (RPC) Service
- Simple TCP/IP Services
- Spooler
- TCP/IP NetBIOS Helper
- Workstation

The default NT registry settings were used, with the exception of tuning 1 parameter. This parameter deals with the SCSI disk controller concurrency. It was changed from the default of 16 to 64 (0x40 hexadecimal). This can be changed by regedit under HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services/aic78xx/Parameters/Device, setting the REG_DWORD 'NumberOfRequests' to 0x40 hexadecimal.

Oracle Parameter Settings

P_BUILD.ORA

```
# p_build.ora
always_anti_join            = hash
db_name                     = tpcd
db_block_buffers            = 2000
shared_pool_size            = 3500000
parallel_max_servers        = 192
#parallel_min_servers       = 80
parallel_min_servers        = 0
parallel_server_idle_time = 2
max_dump_file_size          = 5000
audit_trail                 = FALSE
global_names                = FALSE
commit_point_strength       = 1
dblink_encrypt_login        = true
db_block_size               = 8192
db_file_multiblock_read_count = 16
db_block_lru_latches        = 8
dml_locks                   = 500
processes                   = 1024
sessions                    = 1024
transactions                = 1024
enqueue_resources           = 10240
transactions_per_rollback_segment = 10
distributed_transactions    = 20
log_checkpoint_interval = 9999999999
db_files                    = 1000
open_cursors                = 2000
max_rollback_segments       = 128
sort_area_size              =  1048576
sort_read_fac               = 16
sort_direct_writes          = AUTO
sort_write_buffer_size      = 65536
sort_write_buffers          = 8
hash_area_size              = 20971520
hash_multiblock_io_count    = 1
#####################################
####
compatible                  = 7.3.3
control_files               =
d:\tpcd\dbs\hundgig\cntrl.dbf
background_dump_dest        =
d:\tpcd\log\hundgig
```

user_dump_dest	=	d:\tpcd\log\hundgig

P_LOAD.ORA

```
# p_load.ora
always_anti_join            = hash
db_name                     = tpcd
db_block_buffers            = 20000
shared_pool_size            = 250000000
parallel_server_idle_time = 1
max_dump_file_size          = 5000
audit_trail                 = FALSE
global_names                = FALSE
db_block_size               = 8192
db_file_multiblock_read_count = 32
db_block_lru_latches        = 12
dml_locks                   = 200
processes                   = 512
sessions                    = 512
transactions                = 1024
enqueue_resources           = 20480
transactions_per_rollback_segment = 1
distributed_transactions  = 10
log_checkpoint_interval   = 9999999999
log_buffer                  = 2621440
db_files                    = 1000
open_cursors                = 512
cursor_space_for_time     = FALSE
max_rollback_segments     = 128
sort_area_size              = 1048576
sort_read_fac               = 15
sort_direct_writes          = AUTO
sort_write_buffer_size    = 65536
sort_write_buffers          = 8
optimizer_percent_parallel= 90
hash_area_size              = 20971520
hash_multiblock_io_count  = 2
####################################
#
compatible                  = 7.3.3
control_files       =
d:\tpcd\dbs\hundgig\cntrl.dbf
background_dump_dest  =
d:\tpcd\log\hundgig
optimizer_search_limit    = 7
parallel_max_servers      = 20
parallel_min_servers      = 20
user_dump_dest                  =
d:\tpcd\log\hundgig
v733_plans_enabled        = TRUE
```

P_RUN.ORA

```
# p_run.ora
always_anti_join            = hash
db_name                     = tpcd
parallel_server_idle_time =   1
max_dump_file_size          = 5000
audit_trail                 = FALSE
global_names                = FALSE
db_block_size               = 8192
dml_locks                   = 200
enqueue_resources           = 20480
transactions_per_rollback_segment = 1
log_checkpoint_interval   = 9999999999
log_buffer                  = 2621440
db_files                    = 1000
cursor_space_for_time     = FALSE
max_rollback_segments     = 128
sort_area_size              = 1048576
sort_read_fac               = 15
sort_direct_writes          = AUTO
sort_write_buffer_size    = 65536
sort_write_buffers          = 8
####################################
####
compatible                  = 7.3.3
control_files       =
d:\tpcd\dbs\hundgig\cntrl.dbf
background_dump_dest  =
d:\tpcd\log\hundgig
db_block_buffers            = 18000
db_block_lru_latches        = 4
db_file_multiblock_read_count = 7
hash_area_size              = 20971520
hash_multiblock_io_count  = 7
log_simultaneous_copies   = 8
open_cursors                = 256
optimizer_percent_parallel= 100
optimizer_search_limit    = 7
parallel_max_servers      = 36
parallel_min_servers      = 36
processes                   = 96
sessions                    = 96
transactions                = 128
shared_pool_size            = 90000000
user_dump_dest                  =
d:\tpcd\log\hundgig
v733_plans_enabled        = TRUE
```

P_ANALYZE.ORA

```
# p_analyze.ora
always_anti_join            = hash
db_name                     = tpcd
db_block_buffers            = 2000
shared_pool_size            = 100000000
parallel_max_servers        = 192
parallel_min_servers        = 0
parallel_server_idle_time = 2
max_dump_file_size          = 5000
audit_trail                 = FALSE
global_names                = FALSE
commit_point_strength       = 1
dblink_encrypt_login        = true
db_block_size               = 8192
db_file_multiblock_read_count = 1
db_block_lru_latches        = 8
dml_locks                   = 500
processes                   = 1024
sessions                    = 1024
transactions                = 1024
enqueue_resources           = 10240
transactions_per_rollback_segment = 10
distributed_transactions  = 20
log_checkpoint_interval = 9999999999
db_files                    = 1000
open_cursors                = 2000
max_rollback_segments       = 128
rollback_segments=(r1,r2,r3,r4,r5,r6,r7,r8,r9
,r10,r11,r12,r13,r14,r15,r16,r17,r18,r19,r20,
r21,r22,r23,r24,r25,r26,r27,r28,r29,r30)
sort_area_size              = 1048576
sort_read_fac                   = 16
sort_direct_writes          = AUTO
sort_write_buffer_size      = 65536
sort_write_buffers          = 8
hash_area_size              = 20971520
hash_multiblock_io_count  = 1
###################################
##
compatible                  = 7.3.3
control_files               =
d:\tpcd\dbs\hundgig\cntrl.dbf
background_dump_dest  =
d:\tpcd\log\hundgig
optimizer_search_limit      = 7
user_dump_dest                      =
d:\tpcd\log\hundgig
```

The following batch file was used to increase the stack size allocated by Oracle from 1.0MB to 1.5MB.

RESTACK.BAT

```
editbin /stack: 1572864 oracle73.exe
```

Appendix B: Table Definitions

TPCD100_8_DG.CONF

```
# Configuration file for bumpx.pl
#    TPCD100_8_DG.CONF
# Server configuration:
#    6CPUs, 2GB Memory, 80*4GB disks, 1*2GB disk, 4 controllers,
#    with dual SCSI channels, Windows NT 4.0, Oracle 7.3.3
# TPCD SF=100.
#
*matchon
####################################################################
##
# preprocessing-like directives

%b-preproc

*sql
\svrmgr23 <<!
\set echo on;
\set termout on;
\connect internal;
\{}
\exit;
\!

*load
\sqlldr73 {}

%e-preproc
*matchoff

####################################################################
##

### general configuration parameters

scale_factor = 100
dd_sql_area = d:\orant\rdbms73\admin\
dbs_area = d:\tpcd\dbs\hundgig\
tpcd_sql_area = d:\tpcd\sql\
max_bg = 20
dbcre_max_bg = 20
sccre_max_bg = 20
dapop_max_bg= 8
ixcre_max_bg= 20
anlyz_max_bg= 20
user=tpcd
passwd=tpcd
skip_default_ts=true
skip_ts=ts_undo
skip_mk_ldctlfs=lineitem,orders,customer,parts,partsupp,supplier,nat
ion,region


compatible=7.3

### init.ora parameters

#io_ifile=
io_control_files=d:\tpcd\dbs\hundgig\cntrl.dbf
#do not create ifile=...$parameter{'io_file'}
skip_mk_initoras=include

### database creation parameters

db_maxdatafiles=1020

### system/tablespace parameters

ts_def_area =
g:\tpcd\dbs\hundgig\,h:\tpcd\dbs\hundgig\,i:\tpcd\dbs\hundgig\,&
j:\tpcd\dbs\hundgig\,k:\tpcd\dbs\hundgig\,l:\tpcd\dbs\hundgig\,&
m:\tpcd\dbs\hundgig\,n:\tpcd\dbs\hundgig\

ts_def_#files = 8

ts_sys_first_size=499m
ts_sys_size=499m
ts_sys_area=d:\tpcd\dbs\hundgig\
ts_sys_datafiles=sys.dbf

ts_log_first_size=1900m
ts_log_size=1900m
#ts_log_area=e:\tpcd\dbs\hundgig\
#ts_log_datafiles=log1.dbf,log2.dbf
ts_log_area=none
ts_log_datafiles=\\.\O:,\\.\P:
ts_log_#files_pt=2

ts_undo_first_size=500m
ts_undo_size=500m
#ts_undo_area=d:\tpcd\dbs\hundgig\
#ts_undo_datafiles=ts_undo.dbf
ts_undo_area=none
ts_undo_datafiles=\\.\PHYSICALDRIVEx
ts_undo_#rs=31
ts_undo_rs_storage=(initial 200k next 200k)

# one tablespace per TPCD large object

ts_data_names=ts_lineitem,ts_orders,ts_parts,ts_partsupp,ts_customer
,&
ts_supplier,ts_nation,ts_region
ts_index_names=ts_l_ored,ts_l_pqesod,ts_o_op,ts_ps_pksk,ts_o_clokod,
ts_ps_spsa,&
ts_s_skey,ts_r_rn

ind_indices = l_ored, l_pqesod, o_op, ps_pksk, o_clokod, ps_spsa,
s_skey, r_rn

ts_lineitem_size=9500m
ts_lineitem_storage=(initial 8k next 1899m pctincrease 0)

ts_orders_size=2350m
ts_orders_storage=(initial 8k next 469m pctincrease 0)

ts_parts_size=451m
ts_parts_storage=(initial 8k next 90m pctincrease 0)

ts_partsupp_size=1801m
ts_partsupp_storage=(initial 8k next 360m pctincrease 0)

ts_customer_size=451m
ts_customer_storage=(initial 8k next 90m pctincrease 0)

ts_supplier_size=101m
ts_supplier_storage=(initial 8k next 9m pctincrease 0)
ts_supplier_coalesce_alter=1

ts_nation_#files=1
ts_nation_size=17k
ts_nation_storage=(initial 8k next 8k pctincrease 0)
ts_nation_datafiles=nation.dbf
ts_nation_area=d:\tpcd\dbs\hundgig\
ts_nation_pardeg=1

ts_region_#files=1
ts_region_size=17k
ts_region_storage=(initial 8k next 8k pctincrease 0)
ts_region_datafiles=region.dbf
ts_region_area=d:\tpcd\dbs\hundgig\
ts_region_pardeg=1

ts_l_ored_#files=8
ts_l_ored_size=3175m
ts_l_ored_storage=(initial 8k next 100m maxextents unlimited
pctincrease 0)

ts_l_pqesod_#files=8
ts_l_pqesod_size=3785m
ts_l_pqesod_storage=(initial 8k next 100m maxextents unlimited
pctincrease 0)

ts_o_clokod_#files=8
ts_o_clokod_size=1450m
ts_o_clokod_storage=(initial 8k next 25m maxextents unlimited
pctincrease 0)

ts_o_op_#files=8
ts_o_op_size=1450m
ts_o_op_storage=(initial 8k next 25m maxextents unlimited
pctincrease 0)

ts_ps_pksk_#files=8
ts_ps_pksk_size=948m
ts_ps_pksk_storage=(initial 8k next 20m maxextents unlimited
pctincrease 0)

ts_ps_spsa_#files=8
ts_ps_spsa_size=1128m
ts_ps_spsa_storage=(initial 8k next 20m maxextents unlimited
pctincrease 0)

ts_s_skey_#files=8
ts_s_skey_size=103m
ts_s_skey_storage=(initial 8k next 20m maxextents unlimited
pctincrease 0)

ts_r_rn_#files=1
ts_r_rn_size=17k
ts_r_rn_storage=(initial 8k next 8k maxextents unlimited pctincrease
0)

ts_temp_size=9900m
ts_temp_storage=(initial 8k next 500m maxextents unlimited
pctincrease 0)

### loading  parameters

load_type = delim
#load_no_waits=true
load_field_terminator = '|'
load_tables =
parts,customer,supplier,nation,region,lineitem,orders,partsupp
#load_tables =
lineitem,orders,partsupp,parts,customer,supplier,nation,region
#load_flatfile_area = f:\tpcd\data\hundgig\
load_flatfile_area =  r:\tpcd\data\hundgig\,s:\tpcd\data\hundgig\,&
t:\tpcd\data\hundgig\,u:\tpcd\data\hundgig\,v:\tpcd\data\hundgig\,&
w:\tpcd\data\hundgig\,r:\tpcd\data\hundgig\,s:\tpcd\data\hundgig\,&
t:\tpcd\data\hundgig\,u:\tpcd\data\hundgig\,v:\tpcd\data\hundgig\,&
w:\tpcd\data\hundgig\,r:\tpcd\data\hundgig\,s:\tpcd\data\hundgig\,&
t:\tpcd\data\hundgig\,u:\tpcd\data\hundgig\,v:\tpcd\data\hundgig\,&
w:\tpcd\data\hundgig\,r:\tpcd\data\hundgig\,s:\tpcd\data\hundgig\,&
t:\tpcd\data\hundgig\,u:\tpcd\data\hundgig\,&
v:\tpcd\data\hundgig\,w:\tpcd\data\hundgig\,r:\tpcd\data\hundgig\,&
s:\tpcd\data\hundgig\,t:\tpcd\data\hundgig\,u:\tpcd\data\hundgig\,&
v:\tpcd\data\hundgig\,w:\tpcd\data\hundgig\,r:\tpcd\data\hundgig\,&
s:\tpcd\data\hundgig\,t:\tpcd\data\hundgig\,u:\tpcd\data\hundgig\,&
v:\tpcd\data\hundgig\,r:\tpcd\data\hundgig\,s:\tpcd\data\hundgig\,&
t:\tpcd\data\hundgig\,u:\tpcd\data\hundgig\,v:\tpcd\data\hundgig\

load_controlfile_area = d:\tpcd\ctl\
load_otherfile_area = d:\tpcd\log\hundgig\
load_insert_type = append
load_deg_parallel =40

# table load definitions

#tab_orders_load_degpar =
#tab_lineitem_load_degpar =
#tab_parts_load_degpar =
#tab_customer_load_degpar =
#tab_supplier_load_degpar =
#tab_partsupp_load_degpar =
tab_nation_load_degpar = 1
```

Appendix B

```
tab_region_load_degpar = 1

tab_orders_load_ctlf = orders.ctl
tab_lineitem_load_ctlf = lineitem.ctl
tab_parts_load_ctlf = parts.ctl
tab_partsupp_load_ctlf = partsupp.ctl
tab_customer_load_ctlf = customer.ctl
tab_supplier_load_ctlf = supplier.ctl
tab_nation_load_ctlf = nation.ctl
tab_region_load_ctlf = region.ctl

tab_orders_load_logf = orders#.log
tab_lineitem_load_logf = lineitem#.log
tab_parts_load_logf = parts#.log
tab_partsupp_load_logf = partsupp#.log
tab_customer_load_logf = customer#.log
tab_supplier_load_logf = supplier#.log
tab_region_load_logf = region.log
tab_nation_load_logf = nation.log

tab_orders_load_datf = order.tbl.#
tab_lineitem_load_datf= lineitem.tbl.#
tab_parts_load_datf = part.tbl.#
tab_partsupp_load_datf = partsupp.tbl.#
tab_customer_load_datf = customer.tbl.#
tab_supplier_load_datf = supplier.tbl.#
tab_nation_load_datf = nation.tbl
tab_region_load_datf = region.tbl
tab_nation_load_flatfile_area=w:\tpcd\data\hundgig\
tab_region_load_flatfile_area=w:\tpcd\data\hundgig\

tab_orders_load_filf = alltsdatafiles
tab_lineitem_load_filf = alltsdatafiles
tab_parts_load_filf = alltsdatafiles
tab_partsupp_load_filf = alltsdatafiles
tab_customer_load_filf = alltsdatafiles
tab_supplier_load_filf = alltsdatafiles
tab_nation_load_filf = nation.dbf
tab_region_load_filf = region.dbf

### loading storage parameters

tab_lineitem_storage = (initial 8k next 1899m freelists 8 freelist
groups 9 pctincrease 0)
tab_orders_storage = (initial 8k next 469m freelists 8 freelist
groups 9 pctincrease 0)
tab_parts_storage = (initial 8k next 90m pctincrease 0)
tab_partsupp_storage = (initial 8k next 360m pctincrease 0)
tab_customer_storage = (initial 8k next 90m pctincrease 0)
tab_supplier_storage = (initial 8k next 9m pctincrease 0)
tab_nation_loadextent = 8k
tab_region_loadextent = 8k

# table creation parameters

tab_lineitem_%f = 1
tab_lineitem_%u = 99
tab_lineitem_it=10
tab_lineitem_ts = ts_lineitem
tab_lineitem_pardeg = 60
tab_lineitem_load_parallel=true
tab_lineitem_load_direct=true
tab_lineitem_dealloc_alter=1
tab_lineitem_stor_alter=(next 5m pctincrease 0)

tab_orders_%f = 1
tab_orders_%u = 99
tab_orders_it=10
tab_orders_ts = ts_orders
tab_orders_pardeg = 60
tab_orders_load_parallel=true
tab_orders_load_direct=true
tab_orders_dealloc_alter=1
tab_orders_stor_alter=(next 1m pctincrease 0)

tab_partsupp_%f = 0
tab_partsupp_%u = 99
tab_partsupp_ts =ts_partsupp
tab_partsupp_pardeg = 60
tab_partsupp_load_parallel=true
tab_partsupp_load_direct=true
tab_partsupp_idx_alter=add primary key (ps_partkey,ps_suppkey)
disable
tab_partsupp_pk_alter=enable primary key

tab_parts_%f = 0
tab_parts_%u = 99
tab_parts_ts = ts_parts
tab_parts_pardeg = 60
tab_parts_load_parallel=true
tab_parts_load_direct=true

tab_customer_%f = 0
tab_customer_%u = 99
tab_customer_ts = ts_customer
tab_customer_pardeg = 60
tab_customer_load_parallel=true
tab_customer_load_direct=true

tab_supplier_%f = 0
tab_supplier_%u = 99
tab_supplier_ts = ts_supplier
tab_supplier_pardeg = 60
tab_supplier_load_parallel=true
tab_supplier_load_direct=true

tab_nation_ts = ts_nation
#tab_nation_load_parallel=true
tab_region_ts = ts_region
#tab_region_load_parallel=true

tab_nation_cache = true
tab_region_cache = true

ind_indices =
l_ored,l_pqesod,o_op,ps_pksk,o_clokod,ps_spsa,s_skey,r_rn
ind_unrecoverable=true
ind_pardeg=20

ind_l_ored_table = lineitem
ind_l_ored_tabcols =
l_orderkey,l_returnflag,l_extendedprice,l_discount
ind_l_ored_storage = (initial 20m next 20m freelists 8 freelist
groups 9 maxextents unlimited pctincrease 0)
ind_l_ored_ts = ts_l_ored
ind_l_ored_%f = 2
ind_l_ored_it = 20
#ind_l_ored_pardeg=

ind_l_pqesod_table = lineitem
ind_l_pqesod_tabcols =
l_partkey,l_quantity,l_extendedprice,l_suppkey,l_orderkey,l_discount
ind_l_pqesod_storage = (initial 20m next 20m freelists 8 freelist
groups 9 maxextents unlimited pctincrease 0)
ind_l_pqesod_ts = ts_l_pqesod
ind_l_pqesod_%f = 2
ind_l_pqesod_it = 20
#ind_l_pqesod_pardeg=

ind_o_clokod_table = orders
ind_o_clokod_tabcols = o_clerk,o_orderkey,o_orderdate
ind_o_clokod_storage = (initial 25m next 25m freelists 8 freelist
groups 9 maxextents unlimited pctincrease 0)
ind_o_clokod_ts = ts_o_clokod
ind_o_clokod_%f = 2
ind_o_clokod_it = 20

ind_o_op_table = orders
ind_o_op_tabcols = o_orderkey,o_orderpriority
ind_o_op_storage = (initial 25m next 25m freelists 8 freelist groups
9 maxextents unlimited pctincrease 0)
ind_o_op_ts = ts_o_op
ind_o_op_%f = 2
ind_o_op_it = 20
ind_o_op_unique=1

ind_ps_pksk_table = partsupp
ind_ps_pksk_tabcols = ps_partkey,ps_suppkey
ind_ps_pksk_storage = (initial 20m next 20m maxextents unlimited
pctincrease 0)
ind_ps_pksk_ts = ts_ps_pksk
ind_ps_pksk_%f = 2
ind_ps_pksk_unique=1

ind_ps_spsa_table = partsupp
ind_ps_spsa_tabcols = ps_suppkey, ps_partkey,
ps_supplycost,ps_availqty
ind_ps_spsa_storage = (initial 20m next 20m maxextents unlimited
pctincrease 0)
ind_ps_spsa_ts = ts_ps_spsa
ind_ps_spsa_%f = 2
ind_ps_spsa_unique=1

ind_s_skey_table = supplier
ind_s_skey_tabcols = s_suppkey
ind_s_skey_storage = (initial 20m next 20m maxextents unlimited
pctincrease 0)
ind_s_skey_ts = ts_s_skey
ind_s_skey_%f = 2
ind_s_skey_unique=1

ind_r_rn_table = region
ind_r_rn_tabcols = r_name,r_regionkey
#ind_r_rn_storage = (initial 8k next 8k maxextents unlimited
pctincrease 0)
ind_r_rn_ts = ts_r_rn
#ind_r_rn_%f=
ind_r_rn_pardeg=1
ind_r_rn_unique=1

### Parameters for analyzing objects

#anl_objects=l_ored,l_pqesod,o_op,ps_pksk,o_clokod,ps_spsa,s_skey,r_
rn,lineitem,&
#orders,partsupp,parts,customer,supplier,nation,region

anl_objects
=lineitem,orders,partsupp,parts,customer,supplier,nation,region
anl_historgrams= orders,lineitem,nation,region,parts,partsupp
anl_indexes=l_ored,o_op,l_pqesod,o_clokod,ps_pksk,s_skey,ps_spsa,r_r
n

# Analyze indexes is not in the previous Digital FDR but in the
latest
# According to the manual, it is covered by analyzing tables

anl_l_ored_type = index
anl_l_ored_estimate = sample 100000 rows
anl_o_op_type = index
anl_o_op_estimate = sample 100000 rows
anl_l_pqesod_type = index
anl_l_pqesod_estimate = sample 100000 rows
anl_o_clokod_type = index
anl_o_clokod_estimate = sample 100000 rows
anl_ps_pksk_type = index
anl_ps_pksk_estimate = sample 100000 rows
anl_s_skey_type = index
anl_s_skey_estimate = sample 100000 rows
anl_ps_spsa_type = index
anl_ps_spsa_estimate = sample 100000 rows
anl_r_rn = index
anl_r_rn_estimate = sample 100000 rows

# Analyze tables

anl_lineitem_type = table
anl_lineitem_estimate = sample 100000 rows
anl_orders_type = table
anl_orders_estimate = sample 100000 rows
anl_partsupp_type = table
anl_partsupp_estimate = sample 100000 rows
anl_parts_type = table
anl_parts_estimate = sample 100000 rows
anl_customer_type = table
anl_customer_estimate = sample 100000 rows
anl_supplier_type = table
anl_supplier_estimate = sample 100000 rows
anl_nation_type = table
anl_nation_estimate = sample 100000 rows
anl_region_type = table
anl_region_estimate = sample 100000 rows

# Note: Analyze historgram

anl_lineitem_h_type = table
anl_lineitem_h_estimate = l_orderkey size 250, l_shipdate_size 250,&
 l_commitdate size 250, l_receiptdate size 250, l_suppkey size 250,&
 l_partkey size 250 sample 100000 rows
anl_orders_h_type = table
anl_orders_h_estimate = o_orderkey size 250, o_orderdate size 250,&
 o_clerk size 250, o_custkey size 250 sample 100000 rows
anl_partsupp_h_type = table
anl_partsupp_h_estimate = ps_suppkey size 250, ps_partkey size 250&
 sample 100000 rows
anl_parts_h_type = table
anl_parts_h_estimate = p_name size 250, p_brand size 250,&
```

```
 p_container size 250, p_type size 250, p_size 50 sample 100000 rows
anl_region_h_type = table
anl_region_h_compute =r_name size 5, r_regionkey size 5
anl_nation_h_type = table
anl_nation_h_compute =n_name size 25, n_nationkey size 25
```

BUILD.DAT

```
# preprocessing-like directives

%b-preproc

*sql
\svrmgr23 <<!
\set echo on;
\set termout on;
\connect internal;
\{}
\exit;
\!

*load
\sqlldr73 {}

%e-preproc
%b-dbcre
*bgon=20
######################################################################
###
# Database Creation Phase
*sql
{
shutdown abort
}
*wait
# creating database and initial rollback segment
*sql
{
startup pfile=d:\tpcd\dbs\hundgig\p_build.ora nomount
create database
  controlfile reuse
  logfile '\\.\O:' size 1900m reuse,
          '\\.\P:' size 1900m reuse
  datafile 'd:\tpcd\dbs\hundgig\sys.dbf' size 499m reuse
  maxdatafiles 1020
;

create public rollback segment t_rs1   storage (initial 200k next
200k);

alter rollback segment t_rs1 online;

shutdown
}
*wait
*sql
{
startup pfile=d:\tpcd\dbs\hundgig\p_build.ora
}
*wait
# creating extra rollback segments
*sql
{
create public rollback segment r1 storage (initial 200k next 200k);
create public rollback segment r2 storage (initial 200k next 200k);
create public rollback segment r3 storage (initial 200k next 200k);
create public rollback segment r4 storage (initial 200k next 200k);
create public rollback segment r5 storage (initial 200k next 200k);
create public rollback segment r6 storage (initial 200k next 200k);
create public rollback segment r7 storage (initial 200k next 200k);
create public rollback segment r8 storage (initial 200k next 200k);
create public rollback segment r9 storage (initial 200k next 200k);
create public rollback segment r10 storage (initial 200k next 200k);
create public rollback segment r11 storage (initial 200k next 200k);
create public rollback segment r12 storage (initial 200k next 200k);
create public rollback segment r13 storage (initial 200k next 200k);
create public rollback segment r14 storage (initial 200k next 200k);
create public rollback segment r15 storage (initial 200k next 200k);
create public rollback segment r16 storage (initial 200k next 200k);
create public rollback segment r17 storage (initial 200k next 200k);
create public rollback segment r18 storage (initial 200k next 200k);
create public rollback segment r19 storage (initial 200k next 200k);
create public rollback segment r20 storage (initial 200k next 200k);
create public rollback segment r21 storage (initial 200k next 200k);
create public rollback segment r22 storage (initial 200k next 200k);
create public rollback segment r23 storage (initial 200k next 200k);
create public rollback segment r24 storage (initial 200k next 200k);
create public rollback segment r25 storage (initial 200k next 200k);
create public rollback segment r26 storage (initial 200k next 200k);
create public rollback segment r27 storage (initial 200k next 200k);
create public rollback segment r28 storage (initial 200k next 200k);
create public rollback segment r29 storage (initial 200k next 200k);
create public rollback segment r30 storage (initial 200k next 200k);
}
*wait
# creating extra logfile threads
# building data dictionary
*sql
{
set termout off
set echo off
@d:\orant\rdbms73\admin\catalog.sql
@d:\orant\rdbms73\admin\catparr.sql
@d:\orant\rdbms73\admin\catproc.sql
@d:\orant\rdbms73\admin\utlxplan.sql
}
*wait
*sql
{
shutdown
}
*wait
*sql
{
startup pfile=d:\tpcd\dbs\hundgig\p_build.ora
}
*wait
*wait
*bgoff
%e-dbcre
%b-sccre
*bgon=20
######################################################################
###
# Schema Creation Phase
# creating tpcd user
*sql
{
drop user tpcd cascade;
grant resource,unlimited tablespace,connect
  to tpcd identified by tpcd;
}
########################################

*wait
# creating data tablespaces, datafiles, and tables
# creating tpcd's ts_lineitem tablespace
*sql
{
drop tablespace ts_lineitem including contents;
create tablespace ts_lineitem
  datafile 'g:\tpcd\dbs\hundgig\ts_lineitem1.dbf' size 9500m reuse
  default storage (initial 8k next 1899m pctincrease 0)
;
}
# creating tpcd's ts_orders tablespace
*sql
{
drop tablespace ts_orders including contents;
create tablespace ts_orders
  datafile 'g:\tpcd\dbs\hundgig\ts_orders1.dbf' size 2350m reuse
  default storage (initial 8k next 469m pctincrease 0)
;
}
# creating tpcd's ts_parts tablespace
*sql
{
drop tablespace ts_parts including contents;
create tablespace ts_parts
  datafile 'g:\tpcd\dbs\hundgig\ts_parts1.dbf' size 451m reuse
  default storage (initial 8k next 90m pctincrease 0)
;
}
# creating tpcd's ts_partsupp tablespace
*sql
{
drop tablespace ts_partsupp including contents;
create tablespace ts_partsupp
  datafile 'g:\tpcd\dbs\hundgig\ts_partsupp1.dbf' size 1801m reuse
  default storage (initial 8k next 360m pctincrease 0)
;
}
# creating tpcd's ts_customer tablespace
*sql
{
drop tablespace ts_customer including contents;
create tablespace ts_customer
  datafile 'g:\tpcd\dbs\hundgig\ts_customer1.dbf' size 451m reuse
  default storage (initial 8k next 90m pctincrease 0)
;
}
# creating tpcd's ts_supplier tablespace
*sql
{
drop tablespace ts_supplier including contents;
create tablespace ts_supplier
  datafile 'g:\tpcd\dbs\hundgig\ts_supplier1.dbf' size 101m reuse
  default storage (initial 8k next 9m pctincrease 0)
;
}
# creating tpcd's ts_nation tablespace
*sql
{
drop tablespace ts_nation including contents;
create tablespace ts_nation
  datafile 'd:\tpcd\dbs\hundgig\nation.dbf' size 17k reuse
  default storage (initial 8k next 8k pctincrease 0)
;
}
# creating tpcd's ts_region tablespace
*sql
{
drop tablespace ts_region including contents;
create tablespace ts_region
  datafile 'd:\tpcd\dbs\hundgig\region.dbf' size 17k reuse
  default storage (initial 8k next 8k pctincrease 0)
;
}
# creating tpcd's ts_l_ored tablespace
*sql
{
drop tablespace ts_l_ored including contents;
create tablespace ts_l_ored
  datafile 'g:\tpcd\dbs\hundgig\ts_l_ored1.dbf' size 3175m reuse
  default storage (initial 8k next 100m maxextents unlimited
pctincrease 0)
;
}
# creating tpcd's ts_l_pqesod tablespace
*sql
{
drop tablespace ts_l_pqesod including contents;
create tablespace ts_l_pqesod
  datafile 'g:\tpcd\dbs\hundgig\ts_l_pqesod1.dbf' size 3785m reuse
  default storage (initial 8k next 100m maxextents unlimited
pctincrease 0)
;
}
# creating tpcd's ts_o_op tablespace
*sql
{
drop tablespace ts_o_op including contents;
create tablespace ts_o_op
  datafile 'g:\tpcd\dbs\hundgig\ts_o_op1.dbf' size 1450m reuse
  default storage (initial 8k next 25m maxextents unlimited
pctincrease 0)
;
}
# creating tpcd's ts_ps_pksk tablespace
*sql
{
drop tablespace ts_ps_pksk including contents;
create tablespace ts_ps_pksk
  datafile 'g:\tpcd\dbs\hundgig\ts_ps_pksk1.dbf' size 948m reuse
  default storage (initial 8k next 20m maxextents unlimited
pctincrease 0)
;
}
# creating tpcd's ts_o_clokod tablespace
*sql
{
drop tablespace ts_o_clokod including contents;
create tablespace ts_o_clokod
  datafile 'g:\tpcd\dbs\hundgig\ts_o_clokod1.dbf' size 1450m reuse
  default storage (initial 8k next 25m maxextents unlimited
pctincrease 0)
;
}
# creating tpcd's ts_ps_spsa tablespace
```

Appendix B

```
*sql
{
drop tablespace ts_ps_spsa including contents;
create tablespace ts_ps_spsa
  datafile 'g:\tpcd\dbs\hundgig\ts_ps_spsa1.dbf' size 1128m reuse
  default storage (initial 8k next 20m maxextents unlimited
pctincrease 0)
;
}
# creating tpcd's ts_s_skey tablespace
*sql
{
drop tablespace ts_s_skey including contents;
create tablespace ts_s_skey
  datafile 'g:\tpcd\dbs\hundgig\ts_s_skey1.dbf' size 103m reuse
  default storage (initial 8k next 20m maxextents unlimited
pctincrease 0)
;
}
# creating tpcd's ts_r_rn tablespace
*sql
{
drop tablespace ts_r_rn including contents;
create tablespace ts_r_rn
  datafile 'g:\tpcd\dbs\hundgig\ts_r_rn1.dbf' size 17k reuse
  default storage (initial 8k next 8k maxextents unlimited
pctincrease 0)
;
}
# creating tpcd's ts_temp tablespace
*sql
{
drop tablespace ts_temp including contents;
create tablespace ts_temp temporary
  datafile 'g:\tpcd\dbs\hundgig\ts_temp1.dbf' size 9900m reuse
  default storage (initial 8k next 500m maxextents unlimited
pctincrease 0)
;
}
*wait
*sql
{
connect tpcd/tpcd;
set timing on;
drop table lineitem;
create table lineitem (
    l_shipdate          date ,
    l_orderkey          number ,
    l_discount          number ,
    l_extendedprice     number ,
    l_suppkey           number ,
    l_quantity          number ,
    l_returnflag        char(1) ,
    l_partkey           number ,
    l_linestatus        char(1) ,
    l_tax               number ,
    l_commitdate        date ,
    l_receiptdate       date ,
    l_shipmode          varchar(10) ,
    l_linenumber        number ,
    l_shipinstruct      varchar(25) ,
    l_comment           varchar(44)
)
pctfree 1
pctused 99
initrans 10
tablespace ts_lineitem
storage (initial 8k next 1899m freelists 8 freelist groups 9
pctincrease 0)
parallel (degree 60 )
;
drop table orders;
create table orders (
    o_orderdate         date ,
    o_orderkey          number NOT NULL,
    o_custkey           number NOT NULL,
    o_orderpriority     varchar(15) ,
    o_shippriority      number ,
    o_clerk             varchar(15) ,
    o_orderstatus       char(1) ,
    o_totalprice        number ,
    o_comment           varchar(79)
)
pctfree 1
pctused 99
initrans 10
tablespace ts_orders
storage (initial 8k next 469m freelists 8 freelist groups 9
pctincrease 0)
parallel (degree 60 )
;
drop table partsupp;
create table partsupp (
    ps_partkey          number NOT NULL,
    ps_suppkey          number NOT NULL,
    ps_supplycost       number NOT NULL,
    ps_availqty         number ,
    ps_comment          varchar(199)
)
pctfree 0
pctused 99
tablespace ts_partsupp
storage (initial 8k next 360m pctincrease 0)
parallel (degree 60 )
;
drop table parts;
create table parts (
    p_partkey           number NOT NULL,
    p_type              varchar(25) ,
    p_size              number ,
    p_brand             varchar(10) ,
    p_name              varchar(55) ,
    p_container         varchar(10) ,
    p_mfgr              varchar(25) ,
    p_retailprice       number ,
    p_comment           varchar(23)
)
pctfree 0
pctused 99
tablespace ts_parts
storage (initial 8k next 90m pctincrease 0)
parallel (degree 60 )
;
drop table customer;
create table customer (
    c_custkey           number NOT NULL,
    c_mktsegment        varchar(10) ,
    c_nationkey         number ,
    c_name              varchar(25) ,
    c_address           varchar(40) ,
    c_phone             varchar(15) ,
    c_acctbal           number ,
    c_comment           varchar(117)
)
pctfree 0
pctused 99
tablespace ts_customer
storage (initial 8k next 90m pctincrease 0)
parallel (degree 60 )
;
drop table supplier;
create table supplier (
    s_suppkey           number NOT NULL,
    s_nationkey         number ,
    s_comment           varchar(101) ,
    s_name              varchar(25) ,
    s_address           varchar(40) ,
    s_phone             varchar(15) ,
    s_acctbal           number
)
pctfree 0
pctused 99
tablespace ts_supplier
storage (initial 8k next 9m pctincrease 0)
parallel (degree 60 )
;
drop table nation;
create table nation (
    n_nationkey         number NOT NULL,
    n_name              varchar(25) ,
    n_regionkey         number ,
    n_comment           varchar(152)
)
tablespace ts_nation
cache
;
drop table region;
create table region (
    r_regionkey         number NOT NULL,
    r_name              varchar(25) ,
    r_comment           varchar(152)
)
tablespace ts_region
cache
;
}
*wait
# adding tpcd's ts_lineitem datafiles
*sql
{
alter tablespace ts_lineitem
  add datafile 'h:\tpcd\dbs\hundgig\ts_lineitem2.dbf' size 9500m
reuse;
}
*sql
{
alter tablespace ts_lineitem
  add datafile 'i:\tpcd\dbs\hundgig\ts_lineitem3.dbf' size 9500m
reuse;
}
*sql
{
alter tablespace ts_lineitem
  add datafile 'j:\tpcd\dbs\hundgig\ts_lineitem4.dbf' size 9500m
reuse;
}
*sql
{
alter tablespace ts_lineitem
  add datafile 'k:\tpcd\dbs\hundgig\ts_lineitem5.dbf' size 9500m
reuse;
}
*sql
{
alter tablespace ts_lineitem
  add datafile 'l:\tpcd\dbs\hundgig\ts_lineitem6.dbf' size 9500m
reuse;
}
*sql
{
alter tablespace ts_lineitem
  add datafile 'm:\tpcd\dbs\hundgig\ts_lineitem7.dbf' size 9500m
reuse;
}
*sql
{
alter tablespace ts_lineitem
  add datafile 'n:\tpcd\dbs\hundgig\ts_lineitem8.dbf' size 9500m
reuse;
}
# adding tpcd's ts_orders datafiles
*sql
{
alter tablespace ts_orders
  add datafile 'h:\tpcd\dbs\hundgig\ts_orders2.dbf' size 2350m
reuse;
}
*sql
{
alter tablespace ts_orders
  add datafile 'i:\tpcd\dbs\hundgig\ts_orders3.dbf' size 2350m
reuse;
}
*sql
{
alter tablespace ts_orders
  add datafile 'j:\tpcd\dbs\hundgig\ts_orders4.dbf' size 2350m
reuse;
}
*sql
{
alter tablespace ts_orders
  add datafile 'k:\tpcd\dbs\hundgig\ts_orders5.dbf' size 2350m
reuse;
}
*sql
{
alter tablespace ts_orders
  add datafile 'l:\tpcd\dbs\hundgig\ts_orders6.dbf' size 2350m
reuse;
}
*sql
{
alter tablespace ts_orders
  add datafile 'm:\tpcd\dbs\hundgig\ts_orders7.dbf' size 2350m
reuse;
```

```
}
*sql
{
alter tablespace ts_orders
  add datafile 'n:\tpcd\dbs\hundgig\ts_orders8.dbf' size 2350m
reuse;
}
# adding tpcd's ts_parts datafiles
*sql
{
alter tablespace ts_parts
  add datafile 'h:\tpcd\dbs\hundgig\ts_parts2.dbf' size 451m reuse;
}
*sql
{
alter tablespace ts_parts
  add datafile 'i:\tpcd\dbs\hundgig\ts_parts3.dbf' size 451m reuse;
}
*sql
{
alter tablespace ts_parts
  add datafile 'j:\tpcd\dbs\hundgig\ts_parts4.dbf' size 451m reuse;
}
*sql
{
alter tablespace ts_parts
  add datafile 'k:\tpcd\dbs\hundgig\ts_parts5.dbf' size 451m reuse;
}
*sql
{
alter tablespace ts_parts
  add datafile 'l:\tpcd\dbs\hundgig\ts_parts6.dbf' size 451m reuse;
}
*sql
{
alter tablespace ts_parts
  add datafile 'm:\tpcd\dbs\hundgig\ts_parts7.dbf' size 451m reuse;
}
*sql
{
alter tablespace ts_parts
  add datafile 'n:\tpcd\dbs\hundgig\ts_parts8.dbf' size 451m reuse;
}
# adding tpcd's ts_partsupp datafiles
*sql
{
alter tablespace ts_partsupp
  add datafile 'h:\tpcd\dbs\hundgig\ts_partsupp2.dbf' size 1801m
reuse;
}
*sql
{
alter tablespace ts_partsupp
  add datafile 'i:\tpcd\dbs\hundgig\ts_partsupp3.dbf' size 1801m
reuse;
}
*sql
{
alter tablespace ts_partsupp
  add datafile 'j:\tpcd\dbs\hundgig\ts_partsupp4.dbf' size 1801m
reuse;
}
*sql
{
alter tablespace ts_partsupp
  add datafile 'k:\tpcd\dbs\hundgig\ts_partsupp5.dbf' size 1801m
reuse;
}
*sql
{
alter tablespace ts_partsupp
  add datafile 'l:\tpcd\dbs\hundgig\ts_partsupp6.dbf' size 1801m
reuse;
}
*sql
{
alter tablespace ts_partsupp
  add datafile 'm:\tpcd\dbs\hundgig\ts_partsupp7.dbf' size 1801m
reuse;
}
*sql
{
alter tablespace ts_partsupp
  add datafile 'n:\tpcd\dbs\hundgig\ts_partsupp8.dbf' size 1801m
reuse;
}
# adding tpcd's ts_customer datafiles
*sql
{
alter tablespace ts_customer
  add datafile 'h:\tpcd\dbs\hundgig\ts_customer2.dbf' size 451m
reuse;
}
*sql
{
alter tablespace ts_customer
  add datafile 'i:\tpcd\dbs\hundgig\ts_customer3.dbf' size 451m
reuse;
}
*sql
{
alter tablespace ts_customer
  add datafile 'j:\tpcd\dbs\hundgig\ts_customer4.dbf' size 451m
reuse;
}
*sql
{
alter tablespace ts_customer
  add datafile 'k:\tpcd\dbs\hundgig\ts_customer5.dbf' size 451m
reuse;
}
*sql
{
alter tablespace ts_customer
  add datafile 'l:\tpcd\dbs\hundgig\ts_customer6.dbf' size 451m
reuse;
}
*sql
{
alter tablespace ts_customer
  add datafile 'm:\tpcd\dbs\hundgig\ts_customer7.dbf' size 451m
reuse;
}
*sql
{
alter tablespace ts_customer
  add datafile 'n:\tpcd\dbs\hundgig\ts_customer8.dbf' size 451m
reuse;
}
# adding tpcd's ts_supplier datafiles
*sql
{
alter tablespace ts_supplier
  add datafile 'h:\tpcd\dbs\hundgig\ts_supplier2.dbf' size 101m
reuse;
}
*sql
{
alter tablespace ts_supplier
  add datafile 'i:\tpcd\dbs\hundgig\ts_supplier3.dbf' size 101m
reuse;
}
*sql
{
alter tablespace ts_supplier
  add datafile 'j:\tpcd\dbs\hundgig\ts_supplier4.dbf' size 101m
reuse;
}
*sql
{
alter tablespace ts_supplier
  add datafile 'k:\tpcd\dbs\hundgig\ts_supplier5.dbf' size 101m
reuse;
}
*sql
{
alter tablespace ts_supplier
  add datafile 'l:\tpcd\dbs\hundgig\ts_supplier6.dbf' size 101m
reuse;
}
*sql
{
alter tablespace ts_supplier
  add datafile 'm:\tpcd\dbs\hundgig\ts_supplier7.dbf' size 101m
reuse;
}
*sql
{
alter tablespace ts_supplier
  add datafile 'n:\tpcd\dbs\hundgig\ts_supplier8.dbf' size 101m
reuse;
}
# adding tpcd's ts_l_ored datafiles
*sql
{
alter tablespace ts_l_ored
  add datafile 'h:\tpcd\dbs\hundgig\ts_l_ored2.dbf' size 3175m
reuse;
}
*sql
{
alter tablespace ts_l_ored
  add datafile 'i:\tpcd\dbs\hundgig\ts_l_ored3.dbf' size 3175m
reuse;
}
*sql
{
alter tablespace ts_l_ored
  add datafile 'j:\tpcd\dbs\hundgig\ts_l_ored4.dbf' size 3175m
reuse;
}
*sql
{
alter tablespace ts_l_ored
  add datafile 'k:\tpcd\dbs\hundgig\ts_l_ored5.dbf' size 3175m
reuse;
}
*sql
{
alter tablespace ts_l_ored
  add datafile 'l:\tpcd\dbs\hundgig\ts_l_ored6.dbf' size 3175m
reuse;
}
*sql
{
alter tablespace ts_l_ored
  add datafile 'm:\tpcd\dbs\hundgig\ts_l_ored7.dbf' size 3175m
reuse;
}
*sql
{
alter tablespace ts_l_ored
  add datafile 'n:\tpcd\dbs\hundgig\ts_l_ored8.dbf' size 3175m
reuse;
}
# adding tpcd's ts_l_pqesod datafiles
*sql
{
alter tablespace ts_l_pqesod
  add datafile 'h:\tpcd\dbs\hundgig\ts_l_pqesod2.dbf' size 3785m
reuse;
}
*sql
{
alter tablespace ts_l_pqesod
  add datafile 'i:\tpcd\dbs\hundgig\ts_l_pqesod3.dbf' size 3785m
reuse;
}
*sql
{
alter tablespace ts_l_pqesod
  add datafile 'j:\tpcd\dbs\hundgig\ts_l_pqesod4.dbf' size 3785m
reuse;
}
*sql
{
alter tablespace ts_l_pqesod
  add datafile 'k:\tpcd\dbs\hundgig\ts_l_pqesod5.dbf' size 3785m
reuse;
}
*sql
{
alter tablespace ts_l_pqesod
  add datafile 'l:\tpcd\dbs\hundgig\ts_l_pqesod6.dbf' size 3785m
reuse;
}
*sql
{
alter tablespace ts_l_pqesod
  add datafile 'm:\tpcd\dbs\hundgig\ts_l_pqesod7.dbf' size 3785m
reuse;
}
*sql
{
alter tablespace ts_l_pqesod
```

```
  add datafile 'n:\tpcd\dbs\hundgig\ts_l_pqesod8.dbf' size 3785m
reuse;
}
# adding tpcd's ts_o_op datafiles
*sql
{
alter tablespace ts_o_op
  add datafile 'h:\tpcd\dbs\hundgig\ts_o_op2.dbf' size 1450m reuse;
}
*sql
{
alter tablespace ts_o_op
  add datafile 'i:\tpcd\dbs\hundgig\ts_o_op3.dbf' size 1450m reuse;
}
*sql
{
alter tablespace ts_o_op
  add datafile 'j:\tpcd\dbs\hundgig\ts_o_op4.dbf' size 1450m reuse;
}
*sql
{
alter tablespace ts_o_op
  add datafile 'k:\tpcd\dbs\hundgig\ts_o_op5.dbf' size 1450m reuse;
}
*sql
{
alter tablespace ts_o_op
  add datafile 'l:\tpcd\dbs\hundgig\ts_o_op6.dbf' size 1450m reuse;
}
*sql
{
alter tablespace ts_o_op
  add datafile 'm:\tpcd\dbs\hundgig\ts_o_op7.dbf' size 1450m reuse;
}
*sql
{
alter tablespace ts_o_op
  add datafile 'n:\tpcd\dbs\hundgig\ts_o_op8.dbf' size 1450m reuse;
}
# adding tpcd's ts_ps_pksk datafiles
*sql
{
alter tablespace ts_ps_pksk
  add datafile 'h:\tpcd\dbs\hundgig\ts_ps_pksk2.dbf' size 948m
reuse;
}
*sql
{
alter tablespace ts_ps_pksk
  add datafile 'i:\tpcd\dbs\hundgig\ts_ps_pksk3.dbf' size 948m
reuse;
}
*sql
{
alter tablespace ts_ps_pksk
  add datafile 'j:\tpcd\dbs\hundgig\ts_ps_pksk4.dbf' size 948m
reuse;
}
*sql
{
alter tablespace ts_ps_pksk
  add datafile 'k:\tpcd\dbs\hundgig\ts_ps_pksk5.dbf' size 948m
reuse;
}
*sql
{
alter tablespace ts_ps_pksk
  add datafile 'l:\tpcd\dbs\hundgig\ts_ps_pksk6.dbf' size 948m
reuse;
}
*sql
{
alter tablespace ts_ps_pksk
  add datafile 'm:\tpcd\dbs\hundgig\ts_ps_pksk7.dbf' size 948m
reuse;
}
*sql
{
alter tablespace ts_ps_pksk
  add datafile 'n:\tpcd\dbs\hundgig\ts_ps_pksk8.dbf' size 948m
reuse;
}
# adding tpcd's ts_o_clokod datafiles
*sql
{
alter tablespace ts_o_clokod
  add datafile 'h:\tpcd\dbs\hundgig\ts_o_clokod2.dbf' size 1450m
reuse;
}
*sql
{
alter tablespace ts_o_clokod
  add datafile 'i:\tpcd\dbs\hundgig\ts_o_clokod3.dbf' size 1450m
reuse;
}
*sql
{
alter tablespace ts_o_clokod
  add datafile 'j:\tpcd\dbs\hundgig\ts_o_clokod4.dbf' size 1450m
reuse;
}
*sql
{
alter tablespace ts_o_clokod
  add datafile 'k:\tpcd\dbs\hundgig\ts_o_clokod5.dbf' size 1450m
reuse;
}
*sql
{
alter tablespace ts_o_clokod
  add datafile 'l:\tpcd\dbs\hundgig\ts_o_clokod6.dbf' size 1450m
reuse;
}
*sql
{
alter tablespace ts_o_clokod
  add datafile 'm:\tpcd\dbs\hundgig\ts_o_clokod7.dbf' size 1450m
reuse;
}
*sql
{
alter tablespace ts_o_clokod
  add datafile 'n:\tpcd\dbs\hundgig\ts_o_clokod8.dbf' size 1450m
reuse;
}
# adding tpcd's ts_ps_spsa datafiles
*sql
{
alter tablespace ts_ps_spsa
  add datafile 'h:\tpcd\dbs\hundgig\ts_ps_spsa2.dbf' size 1128m
reuse;
}
*sql
{
alter tablespace ts_ps_spsa
  add datafile 'i:\tpcd\dbs\hundgig\ts_ps_spsa3.dbf' size 1128m
reuse;
}
*sql
{
alter tablespace ts_ps_spsa
  add datafile 'j:\tpcd\dbs\hundgig\ts_ps_spsa4.dbf' size 1128m
reuse;
}
*sql
{
alter tablespace ts_ps_spsa
  add datafile 'k:\tpcd\dbs\hundgig\ts_ps_spsa5.dbf' size 1128m
reuse;
}
*sql
{
alter tablespace ts_ps_spsa
  add datafile 'l:\tpcd\dbs\hundgig\ts_ps_spsa6.dbf' size 1128m
reuse;
}
*sql
{
alter tablespace ts_ps_spsa
  add datafile 'm:\tpcd\dbs\hundgig\ts_ps_spsa7.dbf' size 1128m
reuse;
}
*sql
{
alter tablespace ts_ps_spsa
  add datafile 'n:\tpcd\dbs\hundgig\ts_ps_spsa8.dbf' size 1128m
reuse;
}
# adding tpcd's ts_s_skey datafiles
*sql
{
alter tablespace ts_s_skey
  add datafile 'h:\tpcd\dbs\hundgig\ts_s_skey2.dbf' size 103m reuse;
}
*sql
{
alter tablespace ts_s_skey
  add datafile 'i:\tpcd\dbs\hundgig\ts_s_skey3.dbf' size 103m reuse;
}
*sql
{
alter tablespace ts_s_skey
  add datafile 'j:\tpcd\dbs\hundgig\ts_s_skey4.dbf' size 103m reuse;
}
*sql
{
alter tablespace ts_s_skey
  add datafile 'k:\tpcd\dbs\hundgig\ts_s_skey5.dbf' size 103m reuse;
}
*sql
{
alter tablespace ts_s_skey
  add datafile 'l:\tpcd\dbs\hundgig\ts_s_skey6.dbf' size 103m reuse;
}
*sql
{
alter tablespace ts_s_skey
  add datafile 'm:\tpcd\dbs\hundgig\ts_s_skey7.dbf' size 103m reuse;
}
*sql
{
alter tablespace ts_s_skey
  add datafile 'n:\tpcd\dbs\hundgig\ts_s_skey8.dbf' size 103m reuse;
}
# adding tpcd's ts_temp datafiles
*sql
{
alter tablespace ts_temp
  add datafile 'h:\tpcd\dbs\hundgig\ts_temp2.dbf' size 9900m reuse;
}
*sql
{
alter tablespace ts_temp
  add datafile 'i:\tpcd\dbs\hundgig\ts_temp3.dbf' size 9900m reuse;
}
*sql
{
alter tablespace ts_temp
  add datafile 'j:\tpcd\dbs\hundgig\ts_temp4.dbf' size 9900m reuse;
}
*sql
{
alter tablespace ts_temp
  add datafile 'k:\tpcd\dbs\hundgig\ts_temp5.dbf' size 9900m reuse;
}
*sql
{
alter tablespace ts_temp
  add datafile 'l:\tpcd\dbs\hundgig\ts_temp6.dbf' size 9900m reuse;
}
*sql
{
alter tablespace ts_temp
  add datafile 'm:\tpcd\dbs\hundgig\ts_temp7.dbf' size 9900m reuse;
}
*sql
{
alter tablespace ts_temp
  add datafile 'n:\tpcd\dbs\hundgig\ts_temp8.dbf' size 9900m reuse;
}
*wait
# altering tpcd's temporary tablespace
*sql
{
alter user tpcd temporary tablespace ts_temp;
}
*wait
*wait
*bgoff
%e-sccre
```

LOAD.DAT

```
######################################################################
##
# preprocessing-like directives
```

```
%b-preproc

*sql
\svrmgr23 <<!
\set echo on;
\set termout on;
\connect internal;
\{}
\exit;
\!

*load
\sqlldr73 {}

%e-preproc
%b-shutd
########################################################################
###
# Shutdown Database - All Instances
*time=Shutting down database
*wait
*sql
{
shutdown
}
*wait
*time=Database shutdown
*wait
%e-shutd
%b-start
########################################################################
###
# Startup Database - All Instances
*time=Startup database
*wait
*sql
{
startup pfile=d:\tpcd\dbs\hundgig\p_run.ora
}
*wait
*time=Database started up
*wait
%e-start
%b-scuto
*bgoff
########################################################################
##
# Schema Creation Phase - User and Tables ONLY (no datafiles)
*time=Begin TPCD table cleanup
*wait
*sql
{
connect tpcd/tpcd;
drop index l_ored;
drop index l_pqesod;
drop index o_op;
drop index ps_pksk;
drop index o_clokod;
drop index ps_spsa;
drop index s_skey;
drop index r_rn;
}
*wait
*sql
{
connect tpcd/tpcd;
drop table lineitem;
drop table orders;
drop table partsupp;
drop table parts;
drop table customer;
drop table supplier;
drop table nation;
drop table region;
}
*wait
*sql
{
connect internal;
alter tablespace ts_lineitem coalesce;
alter tablespace ts_orders coalesce;
alter tablespace ts_parts coalesce;
alter tablespace ts_partsupp coalesce;
alter tablespace ts_customer coalesce;
alter tablespace ts_supplier coalesce;
alter tablespace ts_nation coalesce;
alter tablespace ts_region coalesce;
alter tablespace ts_l_ored coalesce;
alter tablespace ts_l_pqesod coalesce;
alter tablespace ts_o_op coalesce;
alter tablespace ts_ps_pksk coalesce;
alter tablespace ts_o_clokod coalesce;
alter tablespace ts_ps_spsa coalesce;
alter tablespace ts_s_skey coalesce;
alter tablespace ts_r_rn coalesce;
alter tablespace ts_temp coalesce;
alter tablespace system coalesce;
connect sys/change_on_install;
@d:\tpcd\sql\orst_cre;
}
*wait
*sql
{
shutdown
}
*wait
*time=Restarting the database for preparation
*wait
*sql
{
startup pfile=d:\tpcd\dbs\hundgig\p_load.ora
}
*wait
*time=End TPCD table cleanup
*wait
*bgoff
*time=Begin TPCD database preparation
*wait
*time=Begin TPCD table creation
*sql
{
connect tpcd/tpcd;
set timing on;
drop table lineitem;
create table lineitem (
    l_shipdate          date ,
    l_orderkey          number ,
    l_discount          number ,
    l_extendedprice     number ,
    l_suppkey           number ,
    l_quantity          number ,
    l_returnflag        char(1) ,
    l_partkey           number ,
    l_linestatus        char(1) ,
    l_tax               number ,
    l_commitdate        date ,
    l_receiptdate       date ,
    l_shipmode          varchar(10) ,
    l_linenumber        number ,
    l_shipinstruct      varchar(25) ,
    l_comment           varchar(44)
)
pctfree 1
pctused 99
initrans 10
tablespace ts_lineitem
storage (initial 8k next 1899m freelists 8 freelist groups 9
pctincrease 0)
parallel (degree 60 )
;
drop table orders;
create table orders (
    o_orderdate         date ,
    o_orderkey          number NOT NULL,
    o_custkey           number NOT NULL,
    o_orderpriority     varchar(15) ,
    o_shippriority      number ,
    o_clerk             varchar(15) ,
    o_orderstatus       char(1) ,
    o_totalprice        number ,
    o_comment           varchar(79)
)
pctfree 1
pctused 99
initrans 10
tablespace ts_orders
storage (initial 8k next 469m freelists 8 freelist groups 9
pctincrease 0)
parallel (degree 60 )
;
drop table partsupp;
create table partsupp (
    ps_partkey          number NOT NULL,
    ps_suppkey          number NOT NULL,
    ps_supplycost       number NOT NULL,
    ps_availqty         number ,
    ps_comment          varchar(199)
)
pctfree 0
pctused 99
tablespace ts_partsupp
storage (initial 8k next 360m pctincrease 0)
parallel (degree 60 )
;
drop table parts;
create table parts (
    p_partkey           number NOT NULL,
    p_type              varchar(25) ,
    p_size              number ,
    p_brand             varchar(10) ,
    p_name              varchar(55) ,
    p_container         varchar(10) ,
    p_mfgr              varchar(25) ,
    p_retailprice       number ,
    p_comment           varchar(23)
)
pctfree 0
pctused 99
tablespace ts_parts
storage (initial 8k next 90m pctincrease 0)
parallel (degree 60 )
;
drop table customer;
create table customer (
    c_custkey           number NOT NULL,
    c_mktsegment        varchar(10) ,
    c_nationkey         number ,
    c_name              varchar(25) ,
    c_address           varchar(40) ,
    c_phone             varchar(15) ,
    c_acctbal           number ,
    c_comment           varchar(117)
)
pctfree 0
pctused 99
tablespace ts_customer
storage (initial 8k next 90m pctincrease 0)
parallel (degree 60 )
;
drop table supplier;
create table supplier (
    s_suppkey           number NOT NULL,
    s_nationkey         number ,
    s_comment           varchar(101) ,
    s_name              varchar(25) ,
    s_address           varchar(40) ,
    s_phone             varchar(15) ,
    s_acctbal           number
)
pctfree 0
pctused 99
tablespace ts_supplier
storage (initial 8k next 9m pctincrease 0)
parallel (degree 60 )
;
drop table nation;
create table nation (
    n_nationkey         number NOT NULL,
    n_name              varchar(25) ,
    n_regionkey         number ,
    n_comment           varchar(152)
)
tablespace ts_nation
cache
;
drop table region;
create table region (
    r_regionkey         number NOT NULL,
    r_name              varchar(25) ,
    r_comment           varchar(152)
)
tablespace ts_region
cache
;
}
```

Appendix B

```
*time=End TPCD table creation
*wait
*bgoff
*wait
*bgoff
%e-scuto
%b-dapop
*bgon=8
#####################################################################
##
# Database Population Phase
*wait
*time=Begin TPCD database load, begin parts
*wait
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts1.log
data=r:\tpcd\data\hundgig\part.tbl.1
FILE=G:\TPCD\DBS\HUNDGIG\TS_PARTS1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts2.log
data=s:\tpcd\data\hundgig\part.tbl.2
FILE=H:\TPCD\DBS\HUNDGIG\TS_PARTS2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts3.log
data=t:\tpcd\data\hundgig\part.tbl.3
FILE=I:\TPCD\DBS\HUNDGIG\TS_PARTS3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts4.log
data=u:\tpcd\data\hundgig\part.tbl.4
FILE=J:\TPCD\DBS\HUNDGIG\TS_PARTS4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts5.log
data=v:\tpcd\data\hundgig\part.tbl.5
FILE=K:\TPCD\DBS\HUNDGIG\TS_PARTS5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts6.log
data=w:\tpcd\data\hundgig\part.tbl.6
FILE=L:\TPCD\DBS\HUNDGIG\TS_PARTS6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts7.log
data=r:\tpcd\data\hundgig\part.tbl.7
FILE=M:\TPCD\DBS\HUNDGIG\TS_PARTS7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts8.log
data=s:\tpcd\data\hundgig\part.tbl.8
FILE=N:\TPCD\DBS\HUNDGIG\TS_PARTS8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts9.log
data=t:\tpcd\data\hundgig\part.tbl.9
FILE=G:\TPCD\DBS\HUNDGIG\TS_PARTS1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts10.log
data=u:\tpcd\data\hundgig\part.tbl.10
FILE=H:\TPCD\DBS\HUNDGIG\TS_PARTS2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts11.log
data=v:\tpcd\data\hundgig\part.tbl.11
FILE=I:\TPCD\DBS\HUNDGIG\TS_PARTS3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts12.log
data=w:\tpcd\data\hundgig\part.tbl.12
FILE=J:\TPCD\DBS\HUNDGIG\TS_PARTS4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts13.log
data=r:\tpcd\data\hundgig\part.tbl.13
FILE=K:\TPCD\DBS\HUNDGIG\TS_PARTS5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts14.log
data=s:\tpcd\data\hundgig\part.tbl.14
FILE=L:\TPCD\DBS\HUNDGIG\TS_PARTS6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts15.log
data=t:\tpcd\data\hundgig\part.tbl.15
FILE=M:\TPCD\DBS\HUNDGIG\TS_PARTS7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts16.log
data=u:\tpcd\data\hundgig\part.tbl.16
FILE=N:\TPCD\DBS\HUNDGIG\TS_PARTS8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts17.log
data=v:\tpcd\data\hundgig\part.tbl.17
FILE=G:\TPCD\DBS\HUNDGIG\TS_PARTS1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts18.log
data=w:\tpcd\data\hundgig\part.tbl.18
FILE=H:\TPCD\DBS\HUNDGIG\TS_PARTS2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts19.log
data=r:\tpcd\data\hundgig\part.tbl.19
FILE=I:\TPCD\DBS\HUNDGIG\TS_PARTS3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts20.log
data=s:\tpcd\data\hundgig\part.tbl.20
FILE=J:\TPCD\DBS\HUNDGIG\TS_PARTS4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts21.log
data=t:\tpcd\data\hundgig\part.tbl.21
FILE=K:\TPCD\DBS\HUNDGIG\TS_PARTS5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts22.log
data=u:\tpcd\data\hundgig\part.tbl.22
FILE=L:\TPCD\DBS\HUNDGIG\TS_PARTS6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts23.log
data=v:\tpcd\data\hundgig\part.tbl.23
FILE=M:\TPCD\DBS\HUNDGIG\TS_PARTS7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts24.log
data=w:\tpcd\data\hundgig\part.tbl.24
FILE=N:\TPCD\DBS\HUNDGIG\TS_PARTS8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts25.log
data=r:\tpcd\data\hundgig\part.tbl.25
FILE=G:\TPCD\DBS\HUNDGIG\TS_PARTS1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts26.log
data=s:\tpcd\data\hundgig\part.tbl.26
FILE=H:\TPCD\DBS\HUNDGIG\TS_PARTS2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts27.log
data=t:\tpcd\data\hundgig\part.tbl.27
FILE=I:\TPCD\DBS\HUNDGIG\TS_PARTS3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts28.log
data=u:\tpcd\data\hundgig\part.tbl.28
FILE=J:\TPCD\DBS\HUNDGIG\TS_PARTS4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts29.log
data=v:\tpcd\data\hundgig\part.tbl.29
FILE=K:\TPCD\DBS\HUNDGIG\TS_PARTS5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts30.log
data=w:\tpcd\data\hundgig\part.tbl.30
FILE=L:\TPCD\DBS\HUNDGIG\TS_PARTS6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts31.log
data=r:\tpcd\data\hundgig\part.tbl.31
FILE=M:\TPCD\DBS\HUNDGIG\TS_PARTS7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts32.log
data=s:\tpcd\data\hundgig\part.tbl.32
FILE=N:\TPCD\DBS\HUNDGIG\TS_PARTS8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts33.log
data=t:\tpcd\data\hundgig\part.tbl.33
FILE=G:\TPCD\DBS\HUNDGIG\TS_PARTS1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts34.log
data=u:\tpcd\data\hundgig\part.tbl.34
FILE=H:\TPCD\DBS\HUNDGIG\TS_PARTS2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts35.log
```

```
data=v:\tpcd\data\hundgig\part.tbl.35
FILE=I:\TPCD\DBS\HUNDGIG\TS_PARTS3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts36.log
data=r:\tpcd\data\hundgig\part.tbl.36
FILE=J:\TPCD\DBS\HUNDGIG\TS_PARTS4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts37.log
data=s:\tpcd\data\hundgig\part.tbl.37
FILE=K:\TPCD\DBS\HUNDGIG\TS_PARTS5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts38.log
data=t:\tpcd\data\hundgig\part.tbl.38
FILE=L:\TPCD\DBS\HUNDGIG\TS_PARTS6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts39.log
data=u:\tpcd\data\hundgig\part.tbl.39
FILE=M:\TPCD\DBS\HUNDGIG\TS_PARTS7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\parts.ctl
log=d:\tpcd\log\hundgig\parts40.log
data=v:\tpcd\data\hundgig\part.tbl.40
FILE=N:\TPCD\DBS\HUNDGIG\TS_PARTS8.DBF direct=true parallel=true
}
*wait
*time=End parts, Begin customer
*wait
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer1.log
data=r:\tpcd\data\hundgig\customer.tbl.1
FILE=G:\TPCD\DBS\HUNDGIG\TS_CUSTOMER1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer2.log
data=s:\tpcd\data\hundgig\customer.tbl.2
FILE=H:\TPCD\DBS\HUNDGIG\TS_CUSTOMER2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer3.log
data=t:\tpcd\data\hundgig\customer.tbl.3
FILE=I:\TPCD\DBS\HUNDGIG\TS_CUSTOMER3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer4.log
data=u:\tpcd\data\hundgig\customer.tbl.4
FILE=J:\TPCD\DBS\HUNDGIG\TS_CUSTOMER4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer5.log
data=v:\tpcd\data\hundgig\customer.tbl.5
FILE=K:\TPCD\DBS\HUNDGIG\TS_CUSTOMER5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer6.log
data=w:\tpcd\data\hundgig\customer.tbl.6
FILE=L:\TPCD\DBS\HUNDGIG\TS_CUSTOMER6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer7.log
data=r:\tpcd\data\hundgig\customer.tbl.7
FILE=M:\TPCD\DBS\HUNDGIG\TS_CUSTOMER7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer8.log
data=s:\tpcd\data\hundgig\customer.tbl.8
FILE=N:\TPCD\DBS\HUNDGIG\TS_CUSTOMER8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer9.log
data=t:\tpcd\data\hundgig\customer.tbl.9
FILE=G:\TPCD\DBS\HUNDGIG\TS_CUSTOMER1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer10.log
data=u:\tpcd\data\hundgig\customer.tbl.10
FILE=H:\TPCD\DBS\HUNDGIG\TS_CUSTOMER2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer11.log
data=v:\tpcd\data\hundgig\customer.tbl.11
FILE=I:\TPCD\DBS\HUNDGIG\TS_CUSTOMER3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer12.log
data=w:\tpcd\data\hundgig\customer.tbl.12
FILE=J:\TPCD\DBS\HUNDGIG\TS_CUSTOMER4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer13.log
data=r:\tpcd\data\hundgig\customer.tbl.13
FILE=K:\TPCD\DBS\HUNDGIG\TS_CUSTOMER5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer14.log
data=s:\tpcd\data\hundgig\customer.tbl.14
FILE=L:\TPCD\DBS\HUNDGIG\TS_CUSTOMER6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer15.log
data=t:\tpcd\data\hundgig\customer.tbl.15
FILE=M:\TPCD\DBS\HUNDGIG\TS_CUSTOMER7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer16.log
data=u:\tpcd\data\hundgig\customer.tbl.16
FILE=N:\TPCD\DBS\HUNDGIG\TS_CUSTOMER8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer17.log
data=v:\tpcd\data\hundgig\customer.tbl.17
FILE=G:\TPCD\DBS\HUNDGIG\TS_CUSTOMER1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer18.log
data=w:\tpcd\data\hundgig\customer.tbl.18
FILE=H:\TPCD\DBS\HUNDGIG\TS_CUSTOMER2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer19.log
data=r:\tpcd\data\hundgig\customer.tbl.19
FILE=I:\TPCD\DBS\HUNDGIG\TS_CUSTOMER3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer20.log
data=s:\tpcd\data\hundgig\customer.tbl.20
FILE=J:\TPCD\DBS\HUNDGIG\TS_CUSTOMER4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer21.log
data=t:\tpcd\data\hundgig\customer.tbl.21
FILE=K:\TPCD\DBS\HUNDGIG\TS_CUSTOMER5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer22.log
data=u:\tpcd\data\hundgig\customer.tbl.22
FILE=L:\TPCD\DBS\HUNDGIG\TS_CUSTOMER6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer23.log
data=v:\tpcd\data\hundgig\customer.tbl.23
FILE=M:\TPCD\DBS\HUNDGIG\TS_CUSTOMER7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer24.log
data=w:\tpcd\data\hundgig\customer.tbl.24
FILE=N:\TPCD\DBS\HUNDGIG\TS_CUSTOMER8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer25.log
data=r:\tpcd\data\hundgig\customer.tbl.25
FILE=G:\TPCD\DBS\HUNDGIG\TS_CUSTOMER1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer26.log
data=s:\tpcd\data\hundgig\customer.tbl.26
FILE=H:\TPCD\DBS\HUNDGIG\TS_CUSTOMER2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer27.log
data=t:\tpcd\data\hundgig\customer.tbl.27
FILE=I:\TPCD\DBS\HUNDGIG\TS_CUSTOMER3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer28.log
data=u:\tpcd\data\hundgig\customer.tbl.28
FILE=J:\TPCD\DBS\HUNDGIG\TS_CUSTOMER4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer29.log
data=v:\tpcd\data\hundgig\customer.tbl.29
FILE=K:\TPCD\DBS\HUNDGIG\TS_CUSTOMER5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer30.log
data=w:\tpcd\data\hundgig\customer.tbl.30
FILE=L:\TPCD\DBS\HUNDGIG\TS_CUSTOMER6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer31.log
```

```
data=r:\tpcd\data\hundgig\customer.tbl.31
FILE=M:\TPCD\DBS\HUNDGIG\TS_CUSTOMER7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer32.log
data=s:\tpcd\data\hundgig\customer.tbl.32
FILE=N:\TPCD\DBS\HUNDGIG\TS_CUSTOMER8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer33.log
data=t:\tpcd\data\hundgig\customer.tbl.33
FILE=G:\TPCD\DBS\HUNDGIG\TS_CUSTOMER1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer34.log
data=u:\tpcd\data\hundgig\customer.tbl.34
FILE=H:\TPCD\DBS\HUNDGIG\TS_CUSTOMER2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer35.log
data=v:\tpcd\data\hundgig\customer.tbl.35
FILE=I:\TPCD\DBS\HUNDGIG\TS_CUSTOMER3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer36.log
data=r:\tpcd\data\hundgig\customer.tbl.36
FILE=J:\TPCD\DBS\HUNDGIG\TS_CUSTOMER4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer37.log
data=s:\tpcd\data\hundgig\customer.tbl.37
FILE=K:\TPCD\DBS\HUNDGIG\TS_CUSTOMER5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer38.log
data=t:\tpcd\data\hundgig\customer.tbl.38
FILE=L:\TPCD\DBS\HUNDGIG\TS_CUSTOMER6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer39.log
data=u:\tpcd\data\hundgig\customer.tbl.39
FILE=M:\TPCD\DBS\HUNDGIG\TS_CUSTOMER7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\customer.ctl
log=d:\tpcd\log\hundgig\customer40.log
data=v:\tpcd\data\hundgig\customer.tbl.40
FILE=N:\TPCD\DBS\HUNDGIG\TS_CUSTOMER8.DBF direct=true parallel=true
}
*wait
*time=End customer, Begin supplier
*wait
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier1.log
data=r:\tpcd\data\hundgig\supplier.tbl.1
FILE=G:\TPCD\DBS\HUNDGIG\TS_SUPPLIER1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier2.log
data=s:\tpcd\data\hundgig\supplier.tbl.2
FILE=H:\TPCD\DBS\HUNDGIG\TS_SUPPLIER2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier3.log
data=t:\tpcd\data\hundgig\supplier.tbl.3
FILE=I:\TPCD\DBS\HUNDGIG\TS_SUPPLIER3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier4.log
data=u:\tpcd\data\hundgig\supplier.tbl.4
FILE=J:\TPCD\DBS\HUNDGIG\TS_SUPPLIER4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier5.log
data=v:\tpcd\data\hundgig\supplier.tbl.5
FILE=K:\TPCD\DBS\HUNDGIG\TS_SUPPLIER5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier6.log
data=w:\tpcd\data\hundgig\supplier.tbl.6
FILE=L:\TPCD\DBS\HUNDGIG\TS_SUPPLIER6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier7.log
data=r:\tpcd\data\hundgig\supplier.tbl.7
FILE=M:\TPCD\DBS\HUNDGIG\TS_SUPPLIER7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier8.log
data=s:\tpcd\data\hundgig\supplier.tbl.8
FILE=N:\TPCD\DBS\HUNDGIG\TS_SUPPLIER8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier9.log
data=t:\tpcd\data\hundgig\supplier.tbl.9
FILE=G:\TPCD\DBS\HUNDGIG\TS_SUPPLIER1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier10.log
data=u:\tpcd\data\hundgig\supplier.tbl.10
FILE=H:\TPCD\DBS\HUNDGIG\TS_SUPPLIER2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier11.log
data=v:\tpcd\data\hundgig\supplier.tbl.11
FILE=I:\TPCD\DBS\HUNDGIG\TS_SUPPLIER3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier12.log
data=w:\tpcd\data\hundgig\supplier.tbl.12
FILE=J:\TPCD\DBS\HUNDGIG\TS_SUPPLIER4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier13.log
data=r:\tpcd\data\hundgig\supplier.tbl.13
FILE=K:\TPCD\DBS\HUNDGIG\TS_SUPPLIER5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier14.log
data=s:\tpcd\data\hundgig\supplier.tbl.14
FILE=L:\TPCD\DBS\HUNDGIG\TS_SUPPLIER6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier15.log
data=t:\tpcd\data\hundgig\supplier.tbl.15
FILE=M:\TPCD\DBS\HUNDGIG\TS_SUPPLIER7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier16.log
data=u:\tpcd\data\hundgig\supplier.tbl.16
FILE=N:\TPCD\DBS\HUNDGIG\TS_SUPPLIER8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier17.log
data=v:\tpcd\data\hundgig\supplier.tbl.17
FILE=G:\TPCD\DBS\HUNDGIG\TS_SUPPLIER1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier18.log
data=w:\tpcd\data\hundgig\supplier.tbl.18
FILE=H:\TPCD\DBS\HUNDGIG\TS_SUPPLIER2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier19.log
data=r:\tpcd\data\hundgig\supplier.tbl.19
FILE=I:\TPCD\DBS\HUNDGIG\TS_SUPPLIER3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier20.log
data=s:\tpcd\data\hundgig\supplier.tbl.20
FILE=J:\TPCD\DBS\HUNDGIG\TS_SUPPLIER4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier21.log
data=t:\tpcd\data\hundgig\supplier.tbl.21
FILE=K:\TPCD\DBS\HUNDGIG\TS_SUPPLIER5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier22.log
data=u:\tpcd\data\hundgig\supplier.tbl.22
FILE=L:\TPCD\DBS\HUNDGIG\TS_SUPPLIER6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier23.log
data=v:\tpcd\data\hundgig\supplier.tbl.23
FILE=M:\TPCD\DBS\HUNDGIG\TS_SUPPLIER7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier24.log
data=w:\tpcd\data\hundgig\supplier.tbl.24
FILE=N:\TPCD\DBS\HUNDGIG\TS_SUPPLIER8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier25.log
data=r:\tpcd\data\hundgig\supplier.tbl.25
FILE=G:\TPCD\DBS\HUNDGIG\TS_SUPPLIER1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier26.log
data=s:\tpcd\data\hundgig\supplier.tbl.26
FILE=H:\TPCD\DBS\HUNDGIG\TS_SUPPLIER2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier27.log
```

```
data=t:\tpcd\data\hundgig\supplier.tbl.27
FILE=I:\TPCD\DBS\HUNDGIG\TS_SUPPLIER3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier28.log
data=u:\tpcd\data\hundgig\supplier.tbl.28
FILE=J:\TPCD\DBS\HUNDGIG\TS_SUPPLIER4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier29.log
data=v:\tpcd\data\hundgig\supplier.tbl.29
FILE=K:\TPCD\DBS\HUNDGIG\TS_SUPPLIER5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier30.log
data=w:\tpcd\data\hundgig\supplier.tbl.30
FILE=L:\TPCD\DBS\HUNDGIG\TS_SUPPLIER6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier31.log
data=r:\tpcd\data\hundgig\supplier.tbl.31
FILE=M:\TPCD\DBS\HUNDGIG\TS_SUPPLIER7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier32.log
data=s:\tpcd\data\hundgig\supplier.tbl.32
FILE=N:\TPCD\DBS\HUNDGIG\TS_SUPPLIER8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier33.log
data=t:\tpcd\data\hundgig\supplier.tbl.33
FILE=G:\TPCD\DBS\HUNDGIG\TS_SUPPLIER1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier34.log
data=u:\tpcd\data\hundgig\supplier.tbl.34
FILE=H:\TPCD\DBS\HUNDGIG\TS_SUPPLIER2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier35.log
data=v:\tpcd\data\hundgig\supplier.tbl.35
FILE=I:\TPCD\DBS\HUNDGIG\TS_SUPPLIER3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier36.log
data=r:\tpcd\data\hundgig\supplier.tbl.36
FILE=J:\TPCD\DBS\HUNDGIG\TS_SUPPLIER4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier37.log
data=s:\tpcd\data\hundgig\supplier.tbl.37
FILE=K:\TPCD\DBS\HUNDGIG\TS_SUPPLIER5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier38.log
data=t:\tpcd\data\hundgig\supplier.tbl.38
FILE=L:\TPCD\DBS\HUNDGIG\TS_SUPPLIER6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier39.log
data=u:\tpcd\data\hundgig\supplier.tbl.39
FILE=M:\TPCD\DBS\HUNDGIG\TS_SUPPLIER7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\supplier.ctl
log=d:\tpcd\log\hundgig\supplier40.log
data=v:\tpcd\data\hundgig\supplier.tbl.40
FILE=N:\TPCD\DBS\HUNDGIG\TS_SUPPLIER8.DBF direct=true parallel=true
}
*wait
*time=End supplier, Begin nation
*wait
*load
{
tpcd/tpcd control=d:\tpcd\ctl\nation.ctl
log=d:\tpcd\log\hundgig\nation.log
data=w:\tpcd\data\hundgig\nation.tbl
FILE=D:\TPCD\DBS\HUNDGIG\NATION.DBF
}
*wait
*time=End nation, Begin region
*wait
*load
{
tpcd/tpcd control=d:\tpcd\ctl\region.ctl
log=d:\tpcd\log\hundgig\region.log
data=w:\tpcd\data\hundgig\region.tbl
FILE=D:\TPCD\DBS\HUNDGIG\REGION.DBF
}
*wait
*time=End region, Begin lineitem
*wait
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem1.log
data=r:\tpcd\data\hundgig\lineitem.tbl.1
FILE=G:\TPCD\DBS\HUNDGIG\TS_LINEITEM1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem2.log
data=s:\tpcd\data\hundgig\lineitem.tbl.2
FILE=H:\TPCD\DBS\HUNDGIG\TS_LINEITEM2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem3.log
data=t:\tpcd\data\hundgig\lineitem.tbl.3
FILE=I:\TPCD\DBS\HUNDGIG\TS_LINEITEM3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem4.log
data=u:\tpcd\data\hundgig\lineitem.tbl.4
FILE=J:\TPCD\DBS\HUNDGIG\TS_LINEITEM4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem5.log
data=v:\tpcd\data\hundgig\lineitem.tbl.5
FILE=K:\TPCD\DBS\HUNDGIG\TS_LINEITEM5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem6.log
data=w:\tpcd\data\hundgig\lineitem.tbl.6
FILE=L:\TPCD\DBS\HUNDGIG\TS_LINEITEM6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem7.log
data=r:\tpcd\data\hundgig\lineitem.tbl.7
FILE=M:\TPCD\DBS\HUNDGIG\TS_LINEITEM7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem8.log
data=s:\tpcd\data\hundgig\lineitem.tbl.8
FILE=N:\TPCD\DBS\HUNDGIG\TS_LINEITEM8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem9.log
data=t:\tpcd\data\hundgig\lineitem.tbl.9
FILE=G:\TPCD\DBS\HUNDGIG\TS_LINEITEM1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem10.log
data=u:\tpcd\data\hundgig\lineitem.tbl.10
FILE=H:\TPCD\DBS\HUNDGIG\TS_LINEITEM2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem11.log
data=v:\tpcd\data\hundgig\lineitem.tbl.11
FILE=I:\TPCD\DBS\HUNDGIG\TS_LINEITEM3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem12.log
data=w:\tpcd\data\hundgig\lineitem.tbl.12
FILE=J:\TPCD\DBS\HUNDGIG\TS_LINEITEM4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem13.log
data=r:\tpcd\data\hundgig\lineitem.tbl.13
FILE=K:\TPCD\DBS\HUNDGIG\TS_LINEITEM5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem14.log
data=s:\tpcd\data\hundgig\lineitem.tbl.14
FILE=L:\TPCD\DBS\HUNDGIG\TS_LINEITEM6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem15.log
data=t:\tpcd\data\hundgig\lineitem.tbl.15
FILE=M:\TPCD\DBS\HUNDGIG\TS_LINEITEM7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem16.log
data=u:\tpcd\data\hundgig\lineitem.tbl.16
FILE=N:\TPCD\DBS\HUNDGIG\TS_LINEITEM8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem17.log
data=v:\tpcd\data\hundgig\lineitem.tbl.17
FILE=G:\TPCD\DBS\HUNDGIG\TS_LINEITEM1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem18.log
data=w:\tpcd\data\hundgig\lineitem.tbl.18
FILE=H:\TPCD\DBS\HUNDGIG\TS_LINEITEM2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem19.log
data=r:\tpcd\data\hundgig\lineitem.tbl.19
FILE=I:\TPCD\DBS\HUNDGIG\TS_LINEITEM3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem20.log
data=s:\tpcd\data\hundgig\lineitem.tbl.20
FILE=J:\TPCD\DBS\HUNDGIG\TS_LINEITEM4.DBF direct=true parallel=true
```

Appendix B

```
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem21.log
data=t:\tpcd\data\hundgig\lineitem.tbl.21
FILE=K:\TPCD\DBS\HUNDGIG\TS_LINEITEM5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem22.log
data=u:\tpcd\data\hundgig\lineitem.tbl.22
FILE=L:\TPCD\DBS\HUNDGIG\TS_LINEITEM6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem23.log
data=v:\tpcd\data\hundgig\lineitem.tbl.23
FILE=M:\TPCD\DBS\HUNDGIG\TS_LINEITEM7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem24.log
data=w:\tpcd\data\hundgig\lineitem.tbl.24
FILE=N:\TPCD\DBS\HUNDGIG\TS_LINEITEM8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem25.log
data=r:\tpcd\data\hundgig\lineitem.tbl.25
FILE=G:\TPCD\DBS\HUNDGIG\TS_LINEITEM1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem26.log
data=s:\tpcd\data\hundgig\lineitem.tbl.26
FILE=H:\TPCD\DBS\HUNDGIG\TS_LINEITEM2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem27.log
data=t:\tpcd\data\hundgig\lineitem.tbl.27
FILE=I:\TPCD\DBS\HUNDGIG\TS_LINEITEM3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem28.log
data=u:\tpcd\data\hundgig\lineitem.tbl.28
FILE=J:\TPCD\DBS\HUNDGIG\TS_LINEITEM4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem29.log
data=v:\tpcd\data\hundgig\lineitem.tbl.29
FILE=K:\TPCD\DBS\HUNDGIG\TS_LINEITEM5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem30.log
data=w:\tpcd\data\hundgig\lineitem.tbl.30
FILE=L:\TPCD\DBS\HUNDGIG\TS_LINEITEM6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem31.log
data=r:\tpcd\data\hundgig\lineitem.tbl.31
FILE=M:\TPCD\DBS\HUNDGIG\TS_LINEITEM7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem32.log
data=s:\tpcd\data\hundgig\lineitem.tbl.32
FILE=N:\TPCD\DBS\HUNDGIG\TS_LINEITEM8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem33.log
data=t:\tpcd\data\hundgig\lineitem.tbl.33
FILE=G:\TPCD\DBS\HUNDGIG\TS_LINEITEM1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem34.log
data=u:\tpcd\data\hundgig\lineitem.tbl.34
FILE=H:\TPCD\DBS\HUNDGIG\TS_LINEITEM2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem35.log
data=v:\tpcd\data\hundgig\lineitem.tbl.35
FILE=I:\TPCD\DBS\HUNDGIG\TS_LINEITEM3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem36.log
data=r:\tpcd\data\hundgig\lineitem.tbl.36
FILE=J:\TPCD\DBS\HUNDGIG\TS_LINEITEM4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem37.log
data=s:\tpcd\data\hundgig\lineitem.tbl.37
FILE=K:\TPCD\DBS\HUNDGIG\TS_LINEITEM5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem38.log
data=t:\tpcd\data\hundgig\lineitem.tbl.38
FILE=L:\TPCD\DBS\HUNDGIG\TS_LINEITEM6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem39.log
data=u:\tpcd\data\hundgig\lineitem.tbl.39
FILE=M:\TPCD\DBS\HUNDGIG\TS_LINEITEM7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\lineitem.ctl
log=d:\tpcd\log\hundgig\lineitem40.log
data=v:\tpcd\data\hundgig\lineitem.tbl.40
FILE=N:\TPCD\DBS\HUNDGIG\TS_LINEITEM8.DBF direct=true parallel=true
}
*wait
*time=End lineitem, Begin orders
*wait
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders1.log
data=r:\tpcd\data\hundgig\order.tbl.1
FILE=G:\TPCD\DBS\HUNDGIG\TS_ORDERS1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders2.log
data=s:\tpcd\data\hundgig\order.tbl.2
FILE=H:\TPCD\DBS\HUNDGIG\TS_ORDERS2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders3.log
data=t:\tpcd\data\hundgig\order.tbl.3
FILE=I:\TPCD\DBS\HUNDGIG\TS_ORDERS3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders4.log
data=u:\tpcd\data\hundgig\order.tbl.4
FILE=J:\TPCD\DBS\HUNDGIG\TS_ORDERS4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders5.log
data=v:\tpcd\data\hundgig\order.tbl.5
FILE=K:\TPCD\DBS\HUNDGIG\TS_ORDERS5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders6.log
data=w:\tpcd\data\hundgig\order.tbl.6
FILE=L:\TPCD\DBS\HUNDGIG\TS_ORDERS6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders7.log
data=r:\tpcd\data\hundgig\order.tbl.7
FILE=M:\TPCD\DBS\HUNDGIG\TS_ORDERS7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders8.log
data=s:\tpcd\data\hundgig\order.tbl.8
FILE=N:\TPCD\DBS\HUNDGIG\TS_ORDERS8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders9.log
data=t:\tpcd\data\hundgig\order.tbl.9
FILE=G:\TPCD\DBS\HUNDGIG\TS_ORDERS1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders10.log
data=u:\tpcd\data\hundgig\order.tbl.10
FILE=H:\TPCD\DBS\HUNDGIG\TS_ORDERS2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders11.log
data=v:\tpcd\data\hundgig\order.tbl.11
FILE=I:\TPCD\DBS\HUNDGIG\TS_ORDERS3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders12.log
data=w:\tpcd\data\hundgig\order.tbl.12
FILE=J:\TPCD\DBS\HUNDGIG\TS_ORDERS4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders13.log
data=r:\tpcd\data\hundgig\order.tbl.13
FILE=K:\TPCD\DBS\HUNDGIG\TS_ORDERS5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders14.log
data=s:\tpcd\data\hundgig\order.tbl.14
FILE=L:\TPCD\DBS\HUNDGIG\TS_ORDERS6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders15.log
data=t:\tpcd\data\hundgig\order.tbl.15
FILE=M:\TPCD\DBS\HUNDGIG\TS_ORDERS7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders16.log
data=u:\tpcd\data\hundgig\order.tbl.16
FILE=N:\TPCD\DBS\HUNDGIG\TS_ORDERS8.DBF direct=true parallel=true
}
```

```
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders17.log
data=v:\tpcd\data\hundgig\order.tbl.17
FILE=G:\TPCD\DBS\HUNDGIG\TS_ORDERS1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders18.log
data=w:\tpcd\data\hundgig\order.tbl.18
FILE=H:\TPCD\DBS\HUNDGIG\TS_ORDERS2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders19.log
data=r:\tpcd\data\hundgig\order.tbl.19
FILE=I:\TPCD\DBS\HUNDGIG\TS_ORDERS3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders20.log
data=s:\tpcd\data\hundgig\order.tbl.20
FILE=J:\TPCD\DBS\HUNDGIG\TS_ORDERS4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders21.log
data=t:\tpcd\data\hundgig\order.tbl.21
FILE=K:\TPCD\DBS\HUNDGIG\TS_ORDERS5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders22.log
data=u:\tpcd\data\hundgig\order.tbl.22
FILE=L:\TPCD\DBS\HUNDGIG\TS_ORDERS6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders23.log
data=v:\tpcd\data\hundgig\order.tbl.23
FILE=M:\TPCD\DBS\HUNDGIG\TS_ORDERS7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders24.log
data=w:\tpcd\data\hundgig\order.tbl.24
FILE=N:\TPCD\DBS\HUNDGIG\TS_ORDERS8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders25.log
data=r:\tpcd\data\hundgig\order.tbl.25
FILE=G:\TPCD\DBS\HUNDGIG\TS_ORDERS1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders26.log
data=s:\tpcd\data\hundgig\order.tbl.26
FILE=H:\TPCD\DBS\HUNDGIG\TS_ORDERS2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders27.log
data=t:\tpcd\data\hundgig\order.tbl.27
FILE=I:\TPCD\DBS\HUNDGIG\TS_ORDERS3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders28.log
data=u:\tpcd\data\hundgig\order.tbl.28
FILE=J:\TPCD\DBS\HUNDGIG\TS_ORDERS4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders29.log
data=v:\tpcd\data\hundgig\order.tbl.29
FILE=K:\TPCD\DBS\HUNDGIG\TS_ORDERS5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders30.log
data=w:\tpcd\data\hundgig\order.tbl.30
FILE=L:\TPCD\DBS\HUNDGIG\TS_ORDERS6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders31.log
data=r:\tpcd\data\hundgig\order.tbl.31
FILE=M:\TPCD\DBS\HUNDGIG\TS_ORDERS7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders32.log
data=s:\tpcd\data\hundgig\order.tbl.32
FILE=N:\TPCD\DBS\HUNDGIG\TS_ORDERS8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders33.log
data=t:\tpcd\data\hundgig\order.tbl.33
FILE=G:\TPCD\DBS\HUNDGIG\TS_ORDERS1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders34.log
data=u:\tpcd\data\hundgig\order.tbl.34
FILE=H:\TPCD\DBS\HUNDGIG\TS_ORDERS2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders35.log
data=v:\tpcd\data\hundgig\order.tbl.35
FILE=I:\TPCD\DBS\HUNDGIG\TS_ORDERS3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders36.log
data=r:\tpcd\data\hundgig\order.tbl.36
FILE=J:\TPCD\DBS\HUNDGIG\TS_ORDERS4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders37.log
data=s:\tpcd\data\hundgig\order.tbl.37
FILE=K:\TPCD\DBS\HUNDGIG\TS_ORDERS5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders38.log
data=t:\tpcd\data\hundgig\order.tbl.38
FILE=L:\TPCD\DBS\HUNDGIG\TS_ORDERS6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders39.log
data=u:\tpcd\data\hundgig\order.tbl.39
FILE=M:\TPCD\DBS\HUNDGIG\TS_ORDERS7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\orders.ctl
log=d:\tpcd\log\hundgig\orders40.log
data=v:\tpcd\data\hundgig\order.tbl.40
FILE=N:\TPCD\DBS\HUNDGIG\TS_ORDERS8.DBF direct=true parallel=true
}
*wait
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp1.log
data=r:\tpcd\data\hundgig\partsupp.tbl.1
FILE=G:\TPCD\DBS\HUNDGIG\TS_PARTSUPP1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp2.log
data=s:\tpcd\data\hundgig\partsupp.tbl.2
FILE=H:\TPCD\DBS\HUNDGIG\TS_PARTSUPP2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp3.log
data=t:\tpcd\data\hundgig\partsupp.tbl.3
FILE=I:\TPCD\DBS\HUNDGIG\TS_PARTSUPP3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp4.log
data=u:\tpcd\data\hundgig\partsupp.tbl.4
FILE=J:\TPCD\DBS\HUNDGIG\TS_PARTSUPP4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp5.log
data=v:\tpcd\data\hundgig\partsupp.tbl.5
FILE=K:\TPCD\DBS\HUNDGIG\TS_PARTSUPP5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp6.log
data=w:\tpcd\data\hundgig\partsupp.tbl.6
FILE=L:\TPCD\DBS\HUNDGIG\TS_PARTSUPP6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp7.log
data=r:\tpcd\data\hundgig\partsupp.tbl.7
FILE=M:\TPCD\DBS\HUNDGIG\TS_PARTSUPP7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp8.log
data=s:\tpcd\data\hundgig\partsupp.tbl.8
FILE=N:\TPCD\DBS\HUNDGIG\TS_PARTSUPP8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp9.log
data=t:\tpcd\data\hundgig\partsupp.tbl.9
FILE=G:\TPCD\DBS\HUNDGIG\TS_PARTSUPP1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp10.log
data=u:\tpcd\data\hundgig\partsupp.tbl.10
FILE=H:\TPCD\DBS\HUNDGIG\TS_PARTSUPP2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp11.log
data=v:\tpcd\data\hundgig\partsupp.tbl.11
FILE=I:\TPCD\DBS\HUNDGIG\TS_PARTSUPP3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp12.log
data=w:\tpcd\data\hundgig\partsupp.tbl.12
FILE=J:\TPCD\DBS\HUNDGIG\TS_PARTSUPP4.DBF direct=true parallel=true
}
*load
{
```

```
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp13.log
data=r:\tpcd\data\hundgig\partsupp.tbl.13
FILE=K:\TPCD\DBS\HUNDGIG\TS_PARTSUPP5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp14.log
data=s:\tpcd\data\hundgig\partsupp.tbl.14
FILE=L:\TPCD\DBS\HUNDGIG\TS_PARTSUPP6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp15.log
data=t:\tpcd\data\hundgig\partsupp.tbl.15
FILE=M:\TPCD\DBS\HUNDGIG\TS_PARTSUPP7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp16.log
data=u:\tpcd\data\hundgig\partsupp.tbl.16
FILE=N:\TPCD\DBS\HUNDGIG\TS_PARTSUPP8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp17.log
data=v:\tpcd\data\hundgig\partsupp.tbl.17
FILE=G:\TPCD\DBS\HUNDGIG\TS_PARTSUPP1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp18.log
data=w:\tpcd\data\hundgig\partsupp.tbl.18
FILE=H:\TPCD\DBS\HUNDGIG\TS_PARTSUPP2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp19.log
data=r:\tpcd\data\hundgig\partsupp.tbl.19
FILE=I:\TPCD\DBS\HUNDGIG\TS_PARTSUPP3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp20.log
data=s:\tpcd\data\hundgig\partsupp.tbl.20
FILE=J:\TPCD\DBS\HUNDGIG\TS_PARTSUPP4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp21.log
data=t:\tpcd\data\hundgig\partsupp.tbl.21
FILE=K:\TPCD\DBS\HUNDGIG\TS_PARTSUPP5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp22.log
data=u:\tpcd\data\hundgig\partsupp.tbl.22
FILE=L:\TPCD\DBS\HUNDGIG\TS_PARTSUPP6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp23.log
data=v:\tpcd\data\hundgig\partsupp.tbl.23
FILE=M:\TPCD\DBS\HUNDGIG\TS_PARTSUPP7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp24.log
data=w:\tpcd\data\hundgig\partsupp.tbl.24
FILE=N:\TPCD\DBS\HUNDGIG\TS_PARTSUPP8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp25.log
data=r:\tpcd\data\hundgig\partsupp.tbl.25
FILE=G:\TPCD\DBS\HUNDGIG\TS_PARTSUPP1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp26.log
data=s:\tpcd\data\hundgig\partsupp.tbl.26
FILE=H:\TPCD\DBS\HUNDGIG\TS_PARTSUPP2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp27.log
data=t:\tpcd\data\hundgig\partsupp.tbl.27
FILE=I:\TPCD\DBS\HUNDGIG\TS_PARTSUPP3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp28.log
data=u:\tpcd\data\hundgig\partsupp.tbl.28
FILE=J:\TPCD\DBS\HUNDGIG\TS_PARTSUPP4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp29.log
data=v:\tpcd\data\hundgig\partsupp.tbl.29
FILE=K:\TPCD\DBS\HUNDGIG\TS_PARTSUPP5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp30.log
data=w:\tpcd\data\hundgig\partsupp.tbl.30
FILE=L:\TPCD\DBS\HUNDGIG\TS_PARTSUPP6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp31.log
data=r:\tpcd\data\hundgig\partsupp.tbl.31
FILE=M:\TPCD\DBS\HUNDGIG\TS_PARTSUPP7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp32.log
data=s:\tpcd\data\hundgig\partsupp.tbl.32
FILE=N:\TPCD\DBS\HUNDGIG\TS_PARTSUPP8.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp33.log
data=t:\tpcd\data\hundgig\partsupp.tbl.33
FILE=G:\TPCD\DBS\HUNDGIG\TS_PARTSUPP1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp34.log
data=u:\tpcd\data\hundgig\partsupp.tbl.34
FILE=H:\TPCD\DBS\HUNDGIG\TS_PARTSUPP2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp35.log
data=v:\tpcd\data\hundgig\partsupp.tbl.35
FILE=I:\TPCD\DBS\HUNDGIG\TS_PARTSUPP3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp36.log
data=r:\tpcd\data\hundgig\partsupp.tbl.36
FILE=J:\TPCD\DBS\HUNDGIG\TS_PARTSUPP4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp37.log
data=s:\tpcd\data\hundgig\partsupp.tbl.37
FILE=K:\TPCD\DBS\HUNDGIG\TS_PARTSUPP5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp38.log
data=t:\tpcd\data\hundgig\partsupp.tbl.38
FILE=L:\TPCD\DBS\HUNDGIG\TS_PARTSUPP6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp39.log
data=u:\tpcd\data\hundgig\partsupp.tbl.39
FILE=M:\TPCD\DBS\HUNDGIG\TS_PARTSUPP7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=d:\tpcd\ctl\partsupp.ctl
log=d:\tpcd\log\hundgig\partsupp40.log
data=v:\tpcd\data\hundgig\partsupp.tbl.40
FILE=N:\TPCD\DBS\HUNDGIG\TS_PARTSUPP8.DBF direct=true parallel=true
}
*wait
*time=End TPCD database load, End partsupp
*wait
*wait
*bgoff
%e-dapop
%b-ixcre
*bgon=20
######################################################################
##
# Index Creation Phase
*time=Begin TPCD analyze
*wait
*sql
{
shutdown
}
*wait
*sql
{
startup pfile=d:\tpcd\dbs\hundgig\p_analyze.ora
}
*wait
*sql
{
connect tpcd/tpcd;
analyze table lineitem estimate statistics sample 100000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze table orders estimate statistics sample 100000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze table partsupp estimate statistics sample 100000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze table parts estimate statistics sample 100000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze table customer estimate statistics sample 100000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze table supplier estimate statistics sample 100000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze table nation estimate statistics sample 100000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze table region estimate statistics sample 100000 rows;
```

```
}
*wait
*time=End table analyze
*wait
*sql
{
shutdown
}
*wait
*sql
{
startup pfile=d:\tpcd\dbs\hundgig\p_index.ora
}
*wait
*time=Begin TPCD index creation
*wait
*sql
{
connect tpcd/tpcd;
drop index l_ored;
create index l_ored
on lineitem (l_orderkey,l_returnflag,l_extendedprice,l_discount)
pctfree 2
initrans 20
tablespace ts_l_ored
storage (initial 20m next 20m freelists 8 freelist groups 9
maxextents unlimited pctincrease 0)
unrecoverable
parallel (degree 20)
;
}
*wait
*time=End index creation: l_ored
*wait
*sql
{
connect tpcd/tpcd;
drop index l_pqesod;
create index l_pqesod
on lineitem
(l_partkey,l_quantity,l_extendedprice,l_suppkey,l_orderkey,l_discoun
t)
pctfree 2
initrans 20
tablespace ts_l_pqesod
storage (initial 20m next 20m freelists 8 freelist groups 9
maxextents unlimited pctincrease 0)
unrecoverable
parallel (degree 20)
;
}
*wait
*time=End index creation: l_pqesod
*wait
*sql
{
connect tpcd/tpcd;
drop index o_op;
create unique index o_op
on orders (o_orderkey,o_orderpriority)
pctfree 2
initrans 20
tablespace ts_o_op
storage (initial 25m next 25m freelists 8 freelist groups 9
maxextents unlimited pctincrease 0)
unrecoverable
parallel (degree 20)
;
}
*wait
*time=End index creation: o_op
*wait
*sql
{
connect tpcd/tpcd;
drop index ps_pksk;
create unique index ps_pksk
on partsupp (ps_partkey,ps_suppkey)
pctfree 2
tablespace ts_ps_pksk
storage (initial 20m next 20m maxextents unlimited pctincrease 0)
unrecoverable
parallel (degree 20)
;
}
*wait
*time=End index creation: ps_pksk
*wait
*sql
{
connect tpcd/tpcd;
drop index o_clokod;
create index o_clokod
on orders (o_clerk,o_orderkey,o_orderdate)
pctfree 2
initrans 20
tablespace ts_o_clokod
storage (initial 25m next 25m freelists 8 freelist groups 9
maxextents unlimited pctincrease 0)
unrecoverable
parallel (degree 20)
;
}
*wait
*time=End index creation: o_clokod
*wait
*sql
{
connect tpcd/tpcd;
drop index ps_spsa;
create unique index ps_spsa
on partsupp (ps_suppkey, ps_partkey, ps_supplycost,ps_availqty)
pctfree 2
tablespace ts_ps_spsa
storage (initial 20m next 20m maxextents unlimited pctincrease 0)
unrecoverable
parallel (degree 20)
;
}
*wait
*time=End index creation: ps_spsa
*wait
*sql
{
connect tpcd/tpcd;
drop index s_skey;
create unique index s_skey
on supplier (s_suppkey)
pctfree 2
tablespace ts_s_skey
storage (initial 20m next 20m maxextents unlimited pctincrease 0)
unrecoverable
parallel (degree 20)
;
}
*wait
*time=End index creation: s_skey
*wait
*sql
{
connect tpcd/tpcd;
drop index r_rn;
create unique index r_rn
on region (r_name,r_regionkey)
tablespace ts_r_rn
unrecoverable
parallel (degree 1)
;
}
*wait
*time=End index creation: r_rn
*wait
*sql
{
connect tpcd/tpcd;
alter table partsupp add primary key (ps_partkey,ps_suppkey)
disable;
alter table partsupp enable primary key;
}
*wait
*time=End TPCD index creation
*wait
*bgoff
%e-ixcre
%b-anlyz
*bgon=20
######################################################################
##
# Analyze Phase
*wait
*time=Begin TPCD analyze
*wait
*sql
{
shutdown
}
*wait
*sql
{
startup pfile=d:\tpcd\dbs\hundgig\p_analyze.ora
}
*wait
*time=Begin index analyze
*wait
*sql
{
connect tpcd/tpcd;
analyze index l_ored estimate statistics sample 100000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze index o_op estimate statistics sample 100000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze index l_pqesod estimate statistics sample 100000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze index o_clokod estimate statistics sample 100000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze index ps_pksk estimate statistics sample 100000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze index s_skey estimate statistics sample 100000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze index ps_spsa estimate statistics sample 100000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze index r_rn estimate statistics sample 100000 rows;
}
*wait
*time=End index analyze
*wait
*time=Begin alter storage parameters
*wait
*sql
{
connect tpcd/tpcd;
alter table lineitem deallocate unused;
alter table orders deallocate unused;
alter table lineitem storage (next 5m pctincrease 0);
alter table orders storage (next 1m pctincrease 0);
connect internal;
alter tablespace ts_supplier coalesce;
alter system switch logfile;
alter system switch logfile;
}
*wait
*sql
{
shutdown
}
*wait
*time=End TPCD database preparation
*wait
*bgoff
%e-anlyz
```

CUSTOMER.CTL

```
---
---  customer.ctl for delimited records
---
```

```
options (direct = true, parallel = true)

unrecoverable

load
into table customer
append
fields terminated by '|'
(
  c_custkey             integer external,
  c_name                ,
  c_address             ,
  c_nationkey           integer external,
  c_phone               ,
  c_acctbal             float external,
  c_mktsegment          ,
  c_comment
)
```

LINEITEM.CTL

```
---
---  lineitem.ctl for delimited records
---

options (direct = true, parallel = true)

unrecoverable

load
into table lineitem
append
fields terminated by '|'
(
  l_orderkey            integer external,
  l_partkey             integer external,
  l_suppkey             integer external,
  l_linenumber          integer external,
  l_quantity            integer external,
  l_extendedprice       float external,
  l_discount            float external,
  l_tax                 float external,
  l_returnflag          ,
  l_linestatus          ,
  l_shipdate            date "yyyy-mm-dd",
  l_commitdate          date "yyyy-mm-dd",
  l_receiptdate         date "yyyy-mm-dd",
  l_shipinstruct        ,
  l_shipmode            ,
  l_comment
)
```

NATION.CTL

```
---
---  nation.ctl for delimited records
---

load
into table nation
append
fields terminated by '|'
(
  n_nationkey           integer external,
  n_name                ,
  n_regionkey           integer external,
  n_comment
)
```

ORDERS.CTL

```
---
---  orders.ctl for delimited records
---

options (direct = true, parallel = true)

unrecoverable

load
into table orders
append
fields terminated by '|'
(
  o_orderkey            integer external,
  o_custkey             integer external,
  o_orderstatus         ,
  o_totalprice          float external,
  o_orderdate           date "yyyy-mm-dd",
  o_orderpriority       ,
  o_clerk               ,
  o_shippriority        integer external,
  o_comment
)
```

PARTS.CTL

```
---
---  parts.ctl for delimited records
---

options (direct = true, parallel = true)

unrecoverable

load
into table parts
append
fields terminated by '|'
(
  p_partkey             integer external,
  p_name                ,
  p_mfgr                ,
  p_brand               ,
  p_type                ,
  p_size                integer external,
  p_container           ,
  p_retailprice         float external,
  p_comment
)
```

PARTSUPP.CTL

```
---
---  partsupp.ctl for delimited records
---

options (direct = true, parallel = true)

unrecoverable

load
into table partsupp
append
fields terminated by '|'
(
  ps_partkey            integer external,
  ps_suppkey            integer external,
  ps_availqty           integer external,
  ps_supplycost         float external,
  ps_comment
)
```

REGION.CTL

```
---
---  region.ctl for delimited records
---

load
into table region
append
fields terminated by '|'
(
  r_regionkey           ,
  r_name                ,
  r_comment
)
```

SUPPLIER.CTL

```
---
---  supplier.ctl for delimited records
---

options (direct = true, parallel = true)

unrecoverable

load
into table supplier
append
fields terminated by '|'
(
  s_suppkey             integer external,
  s_name                ,
  s_address             ,
  s_nationkey           integer external,
  s_phone               ,
  s_acctbal             float external,
  s_comment
)
```

Appendix C: Query Text and Query Output

```
-- using 1123198960 as a seed to the RNG

-- Query 1 Original

SELECT L_RETURNFLAG,
L_LINESTATUS,
SUM(L_QUANTITY) AS SUM_QTY,
SUM(L_EXTENDEDPRICE) AS SUM_BASE_PRICE,
SUM(L_EXTENDEDPRICE * (1 - L_DISCOUNT)) AS SUM_DISC_PRICE,
SUM(L_EXTENDEDPRICE * (1 - L_DISCOUNT) * (1 + L_TAX)) AS SUM_CHARGE,
AVG(L_QUANTITY) AS AVG_QTY,
AVG(L_EXTENDEDPRICE) AS AVG_PRICE,
AVG(L_DISCOUNT) AS AVG_DISC,
COUNT(*) AS COUNT_ORDER
FROM LINEITEM
WHERE L_SHIPDATE <= TO_DATE('1998-12-01','YYYY-MM-DD') - 82
GROUP BY L_RETURNFLAG, L_LINESTATUS
ORDER BY L_RETURNFLAG, L_LINESTATUS

L_RETURNFLAG L_LINESTATUS SUM_QTY          SUM_BASE_PRICE
SUM_DISC_PRICE
SUM_CHARGE          AVG_QTY          AVG_PRICE        AVG_DISC
COUNT_ORDER
A            F            3779273040.00    5666985594905.64
5383626476736.88
5598977619737.66 25.50            38237.70         0.05
148204128.00
N            F            98680891.00      147982302244.32
140578261176.07
146202820808.46  25.50            38244.02         0.05
3869423.00
N            O            7479879826.00    11216118698624.99
10655314230575.39
11081537341726.78 25.50            38238.25         0.05
293321951.00
R            F            3778891288.00    5666363356142.41
5383061406653.68
5598378010496.60 25.50            38235.28         0.05
148197265.00

-- Query 2 (Variant A)

SELECT
S_ACCTBAL,
S_NAME,
N_NAME,
P_PARTKEY,
P_MFGR,
S_ADDRESS,
S_PHONE,
S_COMMENT
FROM PARTS, SUPPLIER, PARTSUPP, NATION, REGION
WHERE P_PARTKEY = PS_PARTKEY
AND S_SUPPKEY = PS_SUPPKEY
AND P_SIZE = 15
AND P_TYPE LIKE '%TIN'
AND S_NATIONKEY = N_NATIONKEY
AND N_REGIONKEY = R_REGIONKEY
AND R_NAME = 'MIDDLE EAST'
AND (P_PARTKEY ,PS_SUPPLYCOST) IN
(SELECT PS_PARTKEY, MIN(PS_SUPPLYCOST)
FROM PARTSUPP, SUPPLIER, NATION, REGION
WHERE S_SUPPKEY = PS_SUPPKEY
AND S_NATIONKEY = N_NATIONKEY
AND N_REGIONKEY = R_REGIONKEY
AND R_NAME = 'MIDDLE EAST'
GROUP BY PS_PARTKEY)
ORDER BY S_ACCTBAL DESC, N_NAME, S_NAME, P_PARTKEY

S_ACCTBAL         S_NAME                      N_NAME
P_PARTKEY         P_MFGR
S_ADDRESS                                   S_PHONE
S_COMMENT
9999.85           Supplier#000245639          JORDAN
10245638.00       Manufacturer#1
lk1mmm10PLQlmMC3hknMgRB3yl                  23-737-431-8921
A4lz0ng2OQ6OllA43i7SA2g1Ck71Miw5 jRS7mxLLmQ6m
9999.72           Supplier#000009450          IRAN
6509437.00        Manufacturer#2
3PSNAxOikORMz SLSmwmh1RB017jxBlA            20-729-897-5564
P52kj041MxRnkC74 m7jB75zQ1PA2wiRLLQRgn
9999.52           Supplier#000264342          IRAN
3264341.00        Manufacturer#1
ANgk2SlC573CPj kkBQ15mS6LA7                 20-813-829-9077
Sm43NNlCy QMNxRz2wzNA6C0nL75RhOPNNQkAkyBy w0wOn6m6Lg7gz
NO3lBwSRjM5LzAl2gy4ggwmny3zACwl
9998.93           Supplier#000151938          IRAN
14401895.00       Manufacturer#3
i0BOkO7M1My04 SyhR5AP56AOl6mnz3S1           20-245-827-7647
kiLyn4MBz071O6kw3CSP6Cl2A3
9998.40           Supplier#000380860          JORDAN
9380859.00        Manufacturer#2
x1P3RhQMSLlhLgQkhS CLM                      23-814-345-7369
B44NRQ0g3l1LBiQ16RmmLni4yh6OxxBgiNNO4 hO700hRjn7A3l

02Qzy7MLNyz0knmB3P63g1Rj57RNRyy3BOnSNM7NyzkhAnhg
9998.20           Supplier#000006343          EGYPT
13506316.00       Manufacturer#4
3g47x12nOkQL6ih2L                           14-617-999-2300
4L0x7774MOjw5C6jhRS2L0l5gkwxxkh3zi5Cxnj2CiMO4BR0B
j2O2wkinwRzglh3Cin6QAB
9998.20           Supplier#000006343          EGYPT
17256291.00       Manufacturer#4
3g47x12nOkQL6ih2L                           14-617-999-2300
4L0x7774MOjw5C6jhRS2L0l5gkwxxkh3zi5Cxnj2CiMO4BR0B
j2O2wkinwRzglh3Cin6QAB
9998.01           Supplier#000614102          EGYPT
11114079.00       Manufacturer#4
 jw5gNwkPzxyS4ikCl                          14-246-804-8213
ziAOSi12OznmMLjgkggAlw03M0MRi577xMwL4yjMN35666hnOmwS6jSNygwCM3R5mMi2
gh5kgggm5jNyRA
9997.97           Supplier#000410275          IRAN
17410274.00       Manufacturer#5
3w51CyQ4h 6OPM103                           20-212-519-5268
kB0iyMmA0QA7Q1xR26hQLmzSPzALORMS0Nzhw1 jkNxkN6O523i lmO0BwMBh3OjxhwM
6ykA4
9997.53           Supplier#000937565          SAUDI ARABIA
10687554.00       Manufacturer#2
21lMy1AhOwSzij721OjLSQm0Q7                  30-555-279-3569
xQwhM4kB6B4BRS6yPkn6MBAhLny7zi1zA0M

-- Query 3

SELECT
L_ORDERKEY,
SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT)) AS REVENUE,
TO_CHAR(O_ORDERDATE, 'YYYY-MM-DD'),
O_SHIPPRIORITY
FROM CUSTOMER, ORDERS, LINEITEM
WHERE C_MKTSEGMENT = 'BUILDING'
AND C_CUSTKEY = O_CUSTKEY
AND L_ORDERKEY = O_ORDERKEY
AND O_ORDERDATE < TO_DATE('1995-03-01','YYYY-MM-DD')
AND L_SHIPDATE > TO_DATE('1995-03-01','YYYY-MM-DD')
GROUP BY L_ORDERKEY, O_ORDERDATE, O_SHIPPRIORITY
ORDER BY REVENUE DESC, O_ORDERDATE

L_ORDERKEY       REVENUE
TO_CHAR(O_ORDERD
O_SHIPPRIORITY
54035522.00      481211.50
1995-02-26
0.00
249762471.00     474337.20
1995-02-11
0.00
365927972.00     474316.43
1995-02-28
0.00
32872099.00      469214.94
1995-01-18
0.00
387682433.00     467789.69
1995-02-03
0.00
505996806.00     466959.23
1995-02-18
0.00
488876289.00     462862.11
1995-02-14
0.00
461314721.00     456178.07
1995-02-06
0.00
156833632.00     453476.91
1995-02-17
0.00
313583264.00     451943.34
1995-02-21
0.00

-- Query 4 (Variant B)

SELECT
O_ORDERPRIORITY,
COUNT(*) AS ORDER_COUNT
FROM ORDERS
WHERE O_ORDERKEY IN (
SELECT DISTINCT O_ORDERKEY
FROM LINEITEM, ORDERS
WHERE L_ORDERKEY = O_ORDERKEY
AND O_ORDERDATE >= TO_DATE('1994-11-01','YYYY-MM-DD')
AND O_ORDERDATE < ADD_MONTHS(TO_DATE('1994-11-01','YYYY-MM-DD'),3)
AND L_COMMITDATE < L_RECEIPTDATE)
GROUP BY O_ORDERPRIORITY
ORDER BY O_ORDERPRIORITY

O_ORDERPRIORITY ORDER_COUNT
1-URGENT        1052491.00
2-HIGH          1052914.00
3-MEDIUM        1053677.00
4-NOT SPECIFIED 1052982.00
5-LOW           1053483.00

-- Query 5

SELECT
N_NAME,
SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT)) AS REVENUE
FROM CUSTOMER, ORDERS, LINEITEM, SUPPLIER, NATION, REGION
WHERE C_CUSTKEY = O_CUSTKEY
AND O_ORDERKEY = L_ORDERKEY
AND L_SUPPKEY = S_SUPPKEY
AND C_NATIONKEY = S_NATIONKEY
AND S_NATIONKEY = N_NATIONKEY
AND N_REGIONKEY = R_REGIONKEY
AND R_NAME = 'AMERICA'
AND O_ORDERDATE >= TO_DATE('1993-01-01','YYYY-MM-DD')
AND O_ORDERDATE < ADD_MONTHS(TO_DATE('1993-01-01','YYYY-MM-DD'),12)
GROUP BY N_NAME
ORDER BY REVENUE DESC

N_NAME                    REVENUE
BRAZIL                    5322487509.97
CANADA                    5311091716.51
ARGENTINA                 5299870641.97
UNITED STATES             5294731739.65
PERU                      5241259074.87

-- Query 6

SELECT
SUM(L_EXTENDEDPRICE * L_DISCOUNT) AS REVENUE
FROM LINEITEM
WHERE L_SHIPDATE >= TO_DATE('1994-01-01','YYYY-MM-DD')
AND L_SHIPDATE < ADD_MONTHS(TO_DATE('1994-01-01','YYYY-MM-DD'),12)
AND L_DISCOUNT BETWEEN 0.02 - 0.01 AND 0.02 + 0.01
AND L_QUANTITY < 25

REVENUE
4469540885.01

-- Query 7 (Original)

SELECT
SUPP_NATION,
CUST_NATION,
YEAR,
SUM(VOLUME) AS REVENUE
FROM
(SELECT
N1.N_NAME AS SUPP_NATION,
```

```
N2.N_NAME AS CUST_NATION,
TO_CHAR(L_SHIPDATE,'YYYY') AS YEAR,
L_EXTENDEDPRICE * (1-L_DISCOUNT) AS VOLUME
FROM SUPPLIER, LINEITEM, ORDERS, CUSTOMER, NATION N1, NATION N2
WHERE S_SUPPKEY = L_SUPPKEY
AND O_ORDERKEY = L_ORDERKEY
AND C_CUSTKEY = O_CUSTKEY
AND S_NATIONKEY = N1.N_NATIONKEY
AND C_NATIONKEY = N2.N_NATIONKEY
AND ((N1.N_NAME = 'INDONESIA' AND N2.N_NAME = 'ALGERIA') OR
(N1.N_NAME = 'ALGERIA' AND N2.N_NAME = 'INDONESIA'))
AND L_SHIPDATE BETWEEN TO_DATE('1995-01-01','YYYY-MM-DD') AND
TO_DATE('1996-12-31','YYYY-MM-DD')
) SHIPPING
GROUP BY SUPP_NATION, CUST_NATION, YEAR
ORDER BY SUPP_NATION, CUST_NATION, YEAR

SUPP_NATION               CUST_NATION
YEAR
REVENUE
ALGERIA                   INDONESIA
1995
5301432755.74
ALGERIA                   INDONESIA
1996
5310955491.23
INDONESIA                 ALGERIA
1995
5251526184.86
INDONESIA                 ALGERIA
1996
5261183565.11

-- Query 8 (Variant B)

SELECT
YEAR,
SUM(DECODE(NATION, 'INDONESIA', VOLUME, 0)) / SUM(VOLUME) AS
MKT_SHARE
FROM
(SELECT
TO_CHAR(O_ORDERDATE,'YYYY') AS YEAR,
L_EXTENDEDPRICE*(1-L_DISCOUNT) AS VOLUME,
N2.N_NAME AS NATION
FROM PARTS, SUPPLIER, LINEITEM, ORDERS, CUSTOMER, NATION N1, NATION
N2,
REGION
WHERE P_PARTKEY = L_PARTKEY
AND S_SUPPKEY = L_SUPPKEY
AND L_ORDERKEY = O_ORDERKEY
AND O_CUSTKEY = C_CUSTKEY
AND C_NATIONKEY = N1.N_NATIONKEY
AND N1.N_REGIONKEY = R_REGIONKEY
AND R_NAME = 'ASIA'
AND S_NATIONKEY = N2.N_NATIONKEY
AND O_ORDERDATE BETWEEN TO_DATE ('1995-01-01','YYYY-MM-DD') AND
TO_DATE('1996-12-31','YYYY-MM-DD')
AND P_TYPE = 'STANDARD ANODIZED TIN'
) ALL_NATIONS
GROUP BY YEAR
ORDER BY YEAR

YEAR
MKT_SHARE
1995
0.04
1996
0.04

-- Query 9 (Original)

SELECT
NATION,
YEAR,
SUM(AMOUNT) AS SUM_PROFIT
FROM
(SELECT
N_NAME AS NATION,
TO_CHAR(O_ORDERDATE,'YYYY') AS YEAR,
L_EXTENDEDPRICE * (1-L_DISCOUNT) - PS_SUPPLYCOST * L_QUANTITY AS
AMOUNT
FROM PARTS, SUPPLIER, LINEITEM, PARTSUPP, ORDERS, NATION
WHERE S_SUPPKEY  = L_SUPPKEY
AND PS_SUPPKEY = L_SUPPKEY
AND PS_PARTKEY = L_PARTKEY
AND P_PARTKEY  = L_PARTKEY
AND O_ORDERKEY = L_ORDERKEY
AND S_NATIONKEY = N_NATIONKEY
AND P_NAME LIKE '%green%'
) PROFIT
GROUP BY NATION, YEAR
ORDER BY NATION, YEAR DESC

NATION
YEAR
SUM_PROFIT
ALGERIA
1998
2703318932.11
ALGERIA
1997
4599040240.91
ALGERIA
1996
4595582892.32
ALGERIA
1995
4579980761.80
ALGERIA
1994
4604453725.11
ALGERIA
1993
4621935974.21
ALGERIA
1992
4596912779.69
ARGENTINA
1998
2654046033.89
ARGENTINA
1997
4547342159.89
ARGENTINA
1996
4556332244.54

-- Query 10

SELECT
C_CUSTKEY,
C_NAME,
SUM(L_EXTENDEDPRICE * (1-L_DISCOUNT)) AS REVENUE,
C_ACCTBAL,
N_NAME,
C_ADDRESS,
C_PHONE,
C_COMMENT
FROM CUSTOMER, ORDERS, LINEITEM, NATION
WHERE C_CUSTKEY = O_CUSTKEY
AND L_ORDERKEY = O_ORDERKEY
AND O_ORDERDATE >= TO_DATE( '1993-10-01' ,'YYYY-MM-DD')
AND O_ORDERDATE < ADD_MONTHS(TO_DATE( '1993-10-01' ,'YYYY-MM-DD'),
3)
AND L_RETURNFLAG = 'R'
AND C_NATIONKEY = N_NATIONKEY
GROUP BY C_CUSTKEY, C_NAME, C_ACCTBAL, C_PHONE, N_NAME, C_ADDRESS,
C_COMMENT
ORDER BY REVENUE DESC

C_CUSTKEY        C_NAME                    REVENUE
C_ACCTBAL
N_NAME                    C_ADDRESS
C_PHONE
C_COMMENT
8851220.00       Customer#008851220        941366.78        972.85
UNITED KINGDOM            3j7LhwnN4ny5zk3nkw2R
33-786-714-2588
x
Qm5Ol66LBjN37Ok1BkCQhB41w13Mk5x4nBLzhMNxwmC1xALPkzx4316iw643wCSO6MmP
ll6NORLLg20z1RShRhB
7010780.00       Customer#007010780        926208.09        8929.15
FRANCE                    l07z2CMNiOyjP 1NNjkwCnLNyQ0j
16-895-601-6077
4374AkgP671QOlP47izhPkOwQ463kiQw2Bkzy7S16zQ1Sw
M5LPziQMx7Nh24mAx4y71m7
4787182.00       Customer#004787182        861557.53        2105.16
BRAZIL                    ljMBClCwiniwzjR5gw5yg5P5PAk6wn
12-829-828-5202
jByOMjklO2hhROzy2gxM mxOSg4yLj
C62ByzlgyPNLiRlCg4Bj35CwjSxnBnCNmiOOMBnNSly0 BSRNPh
7298513.00       Customer#007298513        846898.48        9232.67
UNITED KINGDOM            jyk5MmiL3hNS43SnB2xmnRP
33-688-722-4360
0644wRQ07LBCOAM2kO35jyB52xxP220i0gORjziLPxk4BNBjwh5AwlOLNlS2yiRLOkNB
xxzjiBxRNnN5Bgik03
7213909.00       Customer#007213909        844903.08        -301.02
UNITED STATES             n1SnMRi yAAx5x5S3w750AxlABPNhg
34-331-389-3396
Q0lLLk7MOPy3gknw74POmQgO3CAkRzhLl5kgw0QR3434O
ljAiRlSLnm2OyNm12yQmOB0PPQ 2 LMjBM5kgxhA5PglAh2nCA
8582663.00       Customer#008582663        843800.56        2046.48
MOZAMBIQUE                wBhz5AnnM3y2kQB7LP6gR5jP5zw6n2jhL3ik33BL
26-592-891-4742
RPOBMwQgxk32M4z2 LwAL2NhRCPQ2h12SLxi0QhP6
Qz6nOLSh47Rlg27y56zn76Agz37ShhOmm6C7jRlBRlLAPS1

3kl011CMhnMgBgmxBLghx1
7916320.00       Customer#007916320        843504.40        236.14
INDIA                      6MRyAxih1RP4gjmNNLL2Q7SRRiyRjm01wm6M
18-487-368-7080
5CRLgkRCl5jSP0i4NNC  gLQBlnA4zBQlRm7SmBB7gkmgO4hw4kLm5B1 Bi0C
4810460.00       Customer#004810460        838105.68        3385.29
GERMANY                   kCN7xhOR2jh6wNwLyMw7gS
17-334-590-8410
5lQBLnxSk71C5Qh4nL2Ahk 2 j0jAS1xLxzM7w
RyRSCwO3CP6NQx3LkN75k55nw5wl0z
10419133.00      Customer#010419133        828532.62        482.68
INDONESIA                 Q1BQj  PRn213
19-459-638-5498
RmLmm3BOPMlNy630 4hwn kj4g3QRyy4O6AR5nzQ3PiBh3xlSnw05k5OnNzP23SQNm
CwS LznnmnQ
12333194.00      Customer#012333194        823992.49        3672.27
ETHIOPIA                  PkxwnLQPwi4z2An4iihARBCA ROxMzgy
15-240-183-6591
AggxL55BlA7ik 26wMz701SQL1lA20P6LgmjM11RS466xn2MRP74gPC5xjPz4zgxM
ONh2SPA2lzPwMP4xkLnAL0Q

-- Query 11 (Variant A)

CREATE TABLE PART_VALUE
(PARTNO NUMBER, VALUE NUMBER(20, 2) )
tablespace TS_SUPPLIER
pctfree 0
pctused 99
storage (initial 5m next 5m pctincrease 0)

INSERT INTO PART_VALUE
SELECT
PS_PARTKEY, SUM(PS_SUPPLYCOST*PS_AVAILQTY)
FROM PARTSUPP, SUPPLIER, NATION
WHERE PS_SUPPKEY = S_SUPPKEY
AND S_NATIONKEY = N_NATIONKEY
AND N_NAME = 'INDONESIA'
GROUP BY PS_PARTKEY

CREATE TABLE SUM_PART_VALUE
(TOTAL_VALUE NUMBER(20, 2))
tablespace TS_SUPPLIER
pctfree 0
pctused 99
storage (initial 5m next 5m pctincrease 0)

INSERT INTO SUM_PART_VALUE
SELECT
SUM(PS_SUPPLYCOST*PS_AVAILQTY) * 0.000001
FROM PARTSUPP, SUPPLIER, NATION
WHERE PS_SUPPKEY = S_SUPPKEY
AND S_NATIONKEY = N_NATIONKEY
AND N_NAME = 'INDONESIA'

SELECT
PARTNO, VALUE
FROM PART_VALUE
WHERE VALUE >
(SELECT SUM(TOTAL_VALUE) FROM SUM_PART_VALUE)
ORDER BY VALUE DESC

PARTNO           VALUE
16626224.00      24512598.27
6454730.00       21776855.01
16074544.00      21647166.48
19110448.00      21394803.51
6823071.00       20642547.83
19093515.00      20151385.03
8098754.00       20128096.74
```

```
19401618.00      20104580.74
16882397.00      20047166.35
11688752.00      20029244.46

DROP TABLE PART_VALUE

DROP TABLE SUM_PART_VALUE

-- Query 12 (Variant B)

SELECT
L_SHIPMODE,
SUM(DECODE(O_ORDERPRIORITY,'1-URGENT',1,'2-HIGH',1,0)) AS
HIGH_LINE_COUNT,
SUM(DECODE(O_ORDERPRIORITY,'1-URGENT',0,'2-HIGH',0,1)) AS
LOW_LINE_COUNT
FROM
ORDERS,
LINEITEM
WHERE O_ORDERKEY = L_ORDERKEY
AND L_SHIPMODE IN ('RAIL','REG AIR')
AND L_COMMITDATE < L_RECEIPTDATE
AND L_SHIPDATE < L_COMMITDATE
AND L_RECEIPTDATE >= TO_DATE('1997-01-01','YYYY-MM-DD')
AND L_RECEIPTDATE < ADD_MONTHS(TO_DATE('1997-01-01' ,'YYYY-MM-
DD'),12)
GROUP BY L_SHIPMODE
ORDER BY L_SHIPMODE

L_SHIPMODE HIGH_LINE_COUNT  LOW_LINE_COUNT
RAIL       623181.00        937206.00
REG AIR    624766.00        935366.00

-- Query 13 (Original)

SELECT
YEAR,
SUM(REVENUE) AS REVENUE
FROM
(SELECT
TO_CHAR(O_ORDERDATE,'YYYY') AS YEAR,
L_EXTENDEDPRICE * (1-L_DISCOUNT) AS REVENUE
FROM LINEITEM, ORDERS
WHERE  O_ORDERKEY = L_ORDERKEY
AND O_CLERK = 'Clerk#000000367'
AND L_RETURNFLAG = 'R'
) PERFORMANCE
GROUP BY YEAR
ORDER BY YEAR

YEAR
REVENUE
1992
15108153.81
1993
17867616.79
1994
14193314.87
1995
4014018.83

-- Query 14 (Variant C)

CREATE TABLE ALL_SALES0 (TYPE VARCHAR2(25), AMOUNT NUMBER(20, 2))
tablespace TS_SUPPLIER
pctfree 0
pctused 99
storage (initial 5m next 5m pctincrease 0)

CREATE TABLE SUM_PROMO_SALES0 (PROMO_AMOUNT NUMBER(20, 2))
tablespace TS_SUPPLIER
pctfree 0
pctused 99
storage (initial 5m next 5m pctincrease 0)

CREATE TABLE SUM_ALL_SALES0 (ALL_AMOUNT NUMBER(20, 2))
tablespace TS_SUPPLIER
pctfree 0
pctused 99
storage (initial 5m next 5m pctincrease 0)

INSERT INTO ALL_SALES0
SELECT P_TYPE, SUM(L_EXTENDEDPRICE *(1-L_DISCOUNT))
FROM  LINEITEM, PARTS
WHERE L_PARTKEY = P_PARTKEY
AND L_SHIPDATE >= TO_DATE('1994-11-01','YYYY-MM-DD')
AND L_SHIPDATE < ADD_MONTHS(TO_DATE('1994-11-01','YYYY-MM-DD'), 1)
GROUP BY P_TYPE

INSERT INTO SUM_PROMO_SALES0
SELECT SUM(AMOUNT)
FROM  ALL_SALES0
WHERE TYPE LIKE 'PROMO%'

INSERT INTO SUM_ALL_SALES0
SELECT SUM(AMOUNT)
FROM ALL_SALES0

SELECT 100.00*PROMO_AMOUNT/ALL_AMOUNT AS PROMO_REVENUE
FROM SUM_PROMO_SALES0, SUM_ALL_SALES0

PROMO_REVENUE
16.63

DROP TABLE ALL_SALES0

DROP TABLE SUM_PROMO_SALES0

DROP TABLE SUM_ALL_SALES0

-- Query 15 (Variant B)

CREATE TABLE REVENUE0 (SUPPLIER_NO INTEGER, TOTAL_REVENUE NUMBER(20,
2))
tablespace TS_SUPPLIER
pctfree 0
pctused 99
storage (initial 5m next 5m pctincrease 0)

INSERT INTO REVENUE0
SELECT  L_SUPPKEY, SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT))
FROM  LINEITEM
WHERE L_SHIPDATE >= TO_DATE('1994-11-01','YYYY-MM-DD')
AND L_SHIPDATE <  ADD_MONTHS(TO_DATE('1994-11-01','YYYY-MM-DD'),3)
GROUP BY L_SUPPKEY

SELECT S_SUPPKEY, S_NAME, S_ADDRESS, S_PHONE, TOTAL_REVENUE
FROM  SUPPLIER, REVENUE0
WHERE S_SUPPKEY = SUPPLIER_NO
AND TOTAL_REVENUE = (SELECT MAX(TOTAL_REVENUE) FROM REVENUE0)
ORDER BY S_SUPPKEY

S_SUPPKEY        S_NAME
S_ADDRESS                                S_PHONE
TOTAL_REVENUE
5951.00          Supplier#000005951
A iA3km5R311L4RzAjgj7gOwnAgQ0Q           14-698-491-1588 2391696.22

DROP TABLE REVENUE0

-- Query 16

SELECT
P_BRAND,
P_TYPE,
P_SIZE,
COUNT(DISTINCT PS_SUPPKEY) AS SUPPLIER_CNT
FROM PARTSUPP,PARTS
WHERE P_PARTKEY = PS_PARTKEY
AND P_BRAND <> 'Brand#12'
AND P_TYPE NOT LIKE 'PROMO BRUSHED%'
AND P_SIZE IN (29, 33 ,30 ,46 ,28, 45, 14, 11)
AND PS_SUPPKEY NOT IN (SELECT S_SUPPKEY
FROM SUPPLIER
WHERE S_COMMENT LIKE '%Better Business Bureau%Complaints%')
GROUP BY P_BRAND, P_TYPE, P_SIZE
ORDER BY SUPPLIER_CNT DESC, P_BRAND, P_TYPE, P_SIZE

P_BRAND    P_TYPE                      P_SIZE           SUPPLIER_CNT
Brand#55   ECONOMY BRUSHED COPPER      28.00            636.00
Brand#14   SMALL BRUSHED BRASS         45.00            596.00
Brand#41   LARGE BURNISHED BRASS       33.00            588.00
Brand#31   PROMO PLATED NICKEL         28.00            584.00
Brand#33   PROMO PLATED BRASS          29.00            580.00
Brand#13   STANDARD BRUSHED BRASS      45.00            579.00
Brand#25   PROMO BURNISHED NICKEL      33.00            578.00
Brand#32   STANDARD ANODIZED COPPER    14.00            576.00
Brand#13   SMALL BURNISHED COPPER      33.00            572.00
Brand#34   ECONOMY PLATED STEEL        28.00            571.00

-- Query 17 (Original)

SELECT
SUM(L_EXTENDEDPRICE)/7.0 AS AVG_YEARLY
FROM LINEITEM, PARTS
WHERE P_PARTKEY = L_PARTKEY
AND P_BRAND = 'Brand#12'
AND P_CONTAINER = 'WRAP CAN'
AND L_QUANTITY < (SELECT
0.2* AVG(L1.L_QUANTITY)
FROM LINEITEM L1
WHERE L1.L_PARTKEY = P_PARTKEY)

AVG_YEARLY
31888916.10
```

Appendix D: Seed and Query Substitution Parameters

-- using 1123198960 as a seed to the RNG
-- Query 1 Original

DELTA=82

-- Query 2 (Variant A)

SIZE=15
TYPE=TIN
REGION=MIDDLE EAST

-- Query 3

SEGMENT=BUILDING
DATE=1995-03-01

-- Query 4 (Variant B)

DATE=1994-11-01

-- Query 5

REGION=AMERICA
DATE=1993-01-01

-- Query 6

SHIPDATE=1994-01-01
DISCOUNT=0.02
QUANTITY=25

-- Query 7 (Original)

NATION1=INDONESIA
NATION2=ALGERIA

-- Query 8 (Variant B)

NATION=INDONESIA
REGION=ASIA
TYPE=STANDARD ANODIZED TIN

-- Query 9 (Original)

COLOR=green

-- Query 10

DATE=1993-10-01

-- Query 11 (Variant A)

NATION=INDONESIA
FRACTION=0.000001

-- Query 12 (Variant B)

SHIPMODE1=RAIL
SHIPMODE2=REG AIR
DATE=1997-01-01

-- Query 13 (Original)

CLERK=Clerk#000000367

-- Query 14 (Variant C)

DATE=1994-11-01

-- Query 15 (Variant B)

DATE=1994-11-01

-- Query 16

BRAND=Brand#12
TYPE=PROMO BRUSHED
SIZE IN (29, 33 ,30 ,46 ,28, 45, 14, 11)

-- Query 17 (Original)

BRAND=Brand#12
CONTAINER=WRAP CAN

Appendix E: Implementation Specific Layer/Source Code

The following table lists the source code files used in the benchmark runs.

Implementation–Specific Driver Files

File Name	Description
qexecpl.c	SQL execution engine, OCI version
qexecpl.h	SQL statement execution front-end header file
gettime.c	Get wall clock and CPU time
uf1.pc	TPC-D benchmark UF1 driver, PRO*C version. Also used to restore UF2.
uf2_del.pc	TPC-D benchmark uf2 driver, PRO*C version. Also used to restore UF1.
runuf1.sh	Executes UF1
runuf2.sh	Executes UF2
runpower1.sh	Single-stream TPC-D power test

qexecpl.c

```
/*=================================================================
+
 |       Copyright (c) 1996  Oracle Corp, Redwood Shores, CA
|
 |                    OPEN SYSTEMS PERFORMANCE GROUP
|
 |                         All Rights Reserved
|

+=================================================================+
  | FILENAME
  |    qexecpl.c
  | DESCRIPTION
  |    SQL Execution Engine, OCI version
  | MODIFIED
  |    pswong    04/02/96 - more polishing
  |    pswong    03/25/96 - polish up
  |    pswong          03/06/96 - created

+=================================================================*
/

#include <stdio.h>
#include <string.h>
#include <setjmp.h>
/*#include <sys/param.h>*/
#include <stdlib.h>
#include <errno.h>
#include <math.h>
#include <string.h>
#include <sys/types.h>
#include <time.h>

#include "qexecpl.h"

/* Function Prototypes */

extern double gettime();

/* function prototypes from gen.c */

int get_statement();

/* Declare error handling functions */

void sql_error();
void sql_error2();

/* Other prototypes */

int define_output_variables();
void process_select_list();
void usage();
void SQLinit();
void SQLexec();
void SQLexit();
void *memalloc();
void print_header();
void print_rows();
int OFEN();
void remove_newline();

char logname[UNAME_LEN];  /* username/passwd combo */

double tr_start = 0.0;    /* query start time                    */
double tr_end = 0.0;      /* query end time                      */

double s_tr_start = 0.0;  /* statement start time                */
double s_tr_end = 0.0;    /* statement end time                  */

/* For our purpose of timing, we will treat comments as delimiters
*/
/* for queries.   Thus, we will collect query timings whenever we
*/
/* encounter a comment (of course not for the first comment in a
*/
/* file).
*/

int end_flag = 0;         /* flag to indicate that we have reached
*/
                          /* the end of a query
*/

int stmt_cnt = 0;         /* Number of statements processed.     */
int qry_cnt = 0;          /* Number of query processed.          */

double product = 1.0;     /* cumulative product of query times   */
int rows_ret = 0;         /* the number of rows fetched          */
int num_sel_list = 0;     /* the number of select list item      */

long num_to_fetch = -1;   /* Number of rows to fetch. -1 means fetch
all */

sltype slist[MAX_SEL_LIST];  /* Array for describing Select List
*/
dltype *dlist[MAX_SEL_LIST]; /* Array of ptrs for Defining Select
List   */

char stmt[SQL_LEN];       /* The SQL statement or comment line.  */
char cmnt[81];            /* Buffer to save the comment.         */

FILE *qtemp = stdin;      /* fd for query template               */
FILE *logfile = stdout;   /* log and report files                */
FILE *rep = stdout;

void *defbuf;             /* Buffer pointer for ODEFIN           */
int deflen = 0;           /* Size of data type for ODEFIN        */
int deftype = 1;          /* Oracle type number for ODEFIN       */

time_t tim;               /* To get wall clock time              */
```

Appendix E

```
ldadef tpclda;
csrdef curq;
unsigned long tpchda[256];

/* usage: prints the usage of the program */

void usage() {

  fprintf(stderr,"\nUsage: qexec.ott username/password [q<path name
for query template file>]\n");
  fprintf(stderr,"                  [l<path name for log>] [r<path
name for reports>]\n\n");
  fprintf(stderr,"Options:\n");
  fprintf(stderr,"q<path for query>         : full path name for the
query template file.\n");
  fprintf(stderr,"                            (default is
stdin)\n");
  fprintf(stderr,"l<path name for log>      : full path name for log
files\n");
  fprintf(stderr,"                            (default is
stdout)\n");
  fprintf(stderr,"r<path name for reports>  : full path name for
reports\n");
  fprintf(stderr,"                            (default is
stdout)\n");
  exit(-1);
}



void sql_error(lda, cur)
     ldadef *lda;
     csrdef *cur;
{
  char msg[2048];

  if (cur->rc) {
    oerhms(lda, cur->rc, (text *) msg, 2048);
    fprintf(stderr, "Error: SQL Error encountered in Oracle!\n");
    fputs(msg, stderr);
  }

  /* Rollback just in case */

  orol(lda);

  fprintf(stderr, "Exiting Oracle...\n");
  fflush(stderr);

  SQLexit();

  exit(1);
}

void sql_error2(lda, cur)
     ldadef *lda;
     csrdef *cur;
{

  char msg[2048];

  if (cur->rc) {
    oerhms(lda, cur->rc, (text *) msg, 2048);
    fprintf(stderr, "Error: SQL Error encountered in Oracle!\n");
    fputs(msg, stderr);
  }

  fflush(stderr);
}



void main(argc,argv)
     int argc;
     char *argv[];
{

  int retcode;      /* Return code for get_statement */

  /* Initialize some variables */

  if ((argc > 5) || (argc < 2)) {
    usage();
  }

  /* argv[1] -- User and Password for Database */

  strcpy(logname, argv[1]);

  /* Process  optional parameters */

  argc -= 1;
  argv += 1;

  while(--argc) {
    ++argv;
    switch(argv[0][0]) {
    case 'q':
      if ((qtemp = fopen(++(argv[0]),"r")) == NULL) {
           fprintf(stderr,"Unable to open file '%s'\n", argv[0]);
           fprintf(stderr,"%s: %s\n", argv[0], strerror(errno));
           exit(-1);
      }
      break;
    case 'r':
      if ((rep = fopen(++(argv[0]),"a")) == NULL) {
           fprintf(stderr,"Unable to open file '%s'\n", argv[0]);
           fprintf(stderr,"%s: %s\n", argv[0], strerror(errno));
           exit(-1);
      }
      break;
    case 'l':
      if ((logfile = fopen(++(argv[0]),"a")) == NULL) {
           fprintf(stderr,"Unable to open file '%s'\n", argv[0]);
           fprintf(stderr,"%s: %s\n", argv[0], strerror(errno));
           exit(-1);
      }
      break;
    default:
      fprintf(stderr,"Invalid Option: %c\n", argv[0][0]);
      usage();
      break;
    }
  }

  /* Do some initialization and establish connection with the
database */

  SQLinit();

  /* May want to add some triggering mechanism here */

  time(&tim);
  fprintf(logfile, "Begin Execution at %s\n\n", ctime(&tim));

  /* Get the next statement and start processing it */

  while ((retcode = get_statement()) > 0) {

    switch (retcode) {

      /* If this is a comment, skips it */
    case COMMENT:
      if (end_flag) {
           end_flag = 0;      /* reset query end flag */
           /* save the comment so that we can print it out later on
*/
           strcpy(cmnt, stmt);
           break;
      }
      fprintf(logfile, "%s", stmt);
      fprintf(rep, "%s", stmt);
      break;

      /* if this is a set_row_fetch command */
    case SET_FETCHROW:
      fprintf(logfile,"Setting the number of rows to fetch to:
%ld\n\n",
                num_to_fetch);
      break;

      /* if this is a SQL statement */
    case SQL_STMT:

      /* Executes the query */

      SQLexec();

      s_tr_end = gettime();
      stmt_cnt++;

      /*
      fprintf(logfile,"\nStatement Started at %.2f\n", s_tr_start);
      fprintf(logfile,"Statement Ended at %.2f\n", s_tr_end);
      */
      fprintf(logfile,"Statement Processed in %.2f seconds.\n",
                (s_tr_end - s_tr_start));
      break;

      /* Should never reach here */
    default:
      fprintf(stderr, "Invalid statement type!!\n");
      SQLexit();
      break;
    }
  }

  /* Get Timing for the last query */

  tr_end = gettime();

  time(&tim);
  fprintf(logfile,"\nEnded Executing this Query at %s\n",
ctime(&tim));
  fprintf(logfile,"\nQuery Started at %.2f\n", tr_start);
  fprintf(logfile,"Query Ended at %.2f\n", tr_end);
  fprintf(logfile,"Query Processed in %.2f seconds\n\n",
            (tr_end - tr_start));

  fprintf(rep, "%.2f\n", (tr_end - tr_start));

  fprintf(logfile, "\nSQL statements processed: %d\n", stmt_cnt);
  fprintf(logfile, "Queries processed: %d\n", qry_cnt);

  fflush(rep);
  fflush(logfile);

  /* Close the query template file */

  fclose(qtemp);

  /*  Disconnect from ORACLE. */

  SQLexit();
  exit(0);
}



/* SQLinit(): Perform initialization tasks.
*/
/*            Logs on to Oracle, opens some files and open a cursor
for */
/*            later use.
*/

void SQLinit() {

  int i;

  /* preallocate MAX_PREALLOC members of the dlist array
*/
  /* initializes others to NULL so that we can determine who to free
later */

  for (i=0; i<MAX_SEL_LIST; i++) {
    if (i < MAX_PREALLOC)
      dlist[i] = (dltype *) memalloc (sizeof(dltype));
    else
      dlist[i] = NULL;
  }

  /* Connect to ORACLE.  Program will call sql_error()         */
  /* if an error occurs in connecting to the default database. */

/*  if (orlon(&tpclda, (ub1 *)tpchda, (text *)logname, -1, (text
*)0, -1, 0)) { */
    if ( olog(&tpclda, (ub1 *)tpchda, (text *)logname, -1, (text
*)0, -1, (text *)0, -1,
              (ub4)OCI_LM_NBL)) {
    fprintf (stderr, "Error: Failed to log on\n");
    sql_error(&tpclda, &tpclda);
    exit(-1);
  }
```

```
  printf("\nConnected to ORACLE as user: %s\n\n", logname);

  /* Open a cursor for us to use */

  OOPEN(&tpclda, &curq);

}



/* SQLexec()  Executes the SQL statement.
*/
/*            Parse the SQL statement.
*/
/*            If DDL or DML statements, execute right away.
*/
/*            Else describe and define select list outputs,
*/
/*                 execute and fetch results.
*/

void SQLexec()
{
  int i;

  if (!end_flag) {

    /* Clause 5.3.6.2: QI(i,s) is the time between the first
character */
    /*                 of this query text is submitted and the first
*/
    /*                 character of the next query text is
submitted.  */

    tr_end = gettime();

    if (qry_cnt) {
      time(&tim);
      fprintf(logfile,"\nEnded Executing this Query at %s\n",
ctime(&tim));
      fprintf(logfile,"\nQuery Started at %.2f\n", tr_start);
      fprintf(logfile,"Query Ended at %.2f\n", tr_end);
      fprintf(logfile,"Query Processed in %.2f seconds.\n\n",
                (tr_end - tr_start));

      fprintf(logfile,"----------------------------------------------
----\n\n");

      /* print comments for this query that we have saved */

      fprintf(logfile, "%s\n", cmnt);

      fprintf(rep, "%.2f\n", (tr_end - tr_start));
      fprintf(rep, "%s", cmnt);

      fprintf(logfile,"\nBegan Executing this Query at %s\n",
ctime(&tim));

      /* Let's fflush stuff so that we can see what's going on */

      fflush(logfile);
      fflush(rep);

    }

    tr_start = tr_end;
    qry_cnt++;

    end_flag = 1;
  }

  s_tr_start = gettime();

  /* parse the query, use no defer option to determine the statement
type */
  /* and catch any syntax error before we progress further.
*/

  OPARSE(&tpclda, &curq, stmt, NA, FALSE, VER7);

  /* Prints the query text to the logfile */

  fprintf(logfile, "\n%s\n", stmt);

  /* if this is a DDL or DML statement, execute it right away */
  /* only worries about SELECT statements right now, cannot   */
  /* execute a stored PL/SQL procedure in thie version        */

  if (curq.ft != 4) {
    OEXEC(&tpclda,&curq);
    return;
  }

  /* otherwise, this is a select statement */
  /* Describe and define output variables  */

  num_sel_list = define_output_variables();

  /* Executes the query and fetches the rows */

  (void) process_select_list(num_sel_list);


  /* Need to get the number of rows fetched first   */
  /* since the following statments will screw it up */

  rows_ret = curq.rpc;

  /* To control memory usage, let's free up the extra dlist entries
*/
  /* that we have allocated.
*/

  i=MAX_PREALLOC;

  while(dlist[i] != NULL) {
           free(dlist[i]);
           dlist[i++] = NULL;
  }

  /* reset set_fetchrows */

  num_to_fetch = -1;

}

void SQLexit() {

  int i;

  oclose(&curq);
  ologof(&tpclda);

  /* free all memory */

  for (i=0; i<MAX_SEL_LIST; i++) {
    if (dlist[i] != NULL) {
      free(dlist[i]);
           dlist[i] = NULL;
         }
  }

  /* Flush all output */

  fflush(rep);
  fflush(logfile);

}


/* define_output_variables(): Describe and define select-list items
for */
/*                            a query statement.
*/
/*                            Returns the number of select-list
items   */
/*                            for this query.
*/

int define_output_variables()
{

  int i;

  for (i=0; i<MAX_SEL_LIST; i++) {

    slist[i].buflen = MAX_COLNAME_SIZE;

    if (odescr(&curq, POS(i), (sb4 *) &slist[i].dbsize,
                       (sb2 *) &slist[i].dbtype, (sb1 *)
&(slist[i].buf[0]),
                       (sb4 *) &slist[i].buflen, (sb4 *)
&slist[i].dsize,
                       (sb2 *) &slist[i].precision, (sb2 *)
&slist[i].scale,
                       (sb2 *) &slist[i].nullok)) {
      if (curq.rc == END_OF_LIST)
           break;
      else {
           sql_error(&tpclda, &curq);
           return -1;
      }
    }

    /* For formatting purpose, remove trailing blanks in select-list
name. */

    if (slist[i].buflen < MAX_COLNAME_SIZE)
      (slist[i].buf)[slist[i].buflen] = '\0';

    /* Well, we need to allocate for entries for dlist */

    if (i >= MAX_PREALLOC)
      dlist[i] = (dltype *) memalloc(sizeof(dltype));

    /* Let's check the sizes and types for this select list item */

    switch (slist[i].dbtype) {

    case NUM_TYPE:

      /* see if it is an integer or a floating point number */

      slist[i].dbsize = NUMWIDTH;

      /* The odescr will not give a good estimate to the scale if */
      /* no scale was given in the Oracle table definition.       */

#ifdef HAVE_SCALE
      if (slist[i].scale != 0) {
           defbuf = (double *) dlist[i]->fbuf;
           deflen = FLT;
           deftype = FLT_TYPE;
           slist[i].dbtype = FLT_TYPE;
      } else {
           defbuf = (int *) dlist[i]->ibuf;
           deflen = INT;
           deftype = INT_TYPE;
           slist[i].dbtype = INT_TYPE;
      }
#else
      defbuf = (double *) dlist[i]->fbuf;
      deflen = FLT;
      deftype = FLT_TYPE;
      slist[i].dbtype = FLT_TYPE;
#endif /* HAVE_SCALE */

      break;

    default:

      /* default is character string */

      defbuf = (char **) dlist[i]->sbuf;
      deflen = MAX_STR_LEN;
      deftype = STR_TYPE;
      break;

    }

    /* Define the column */

    ODEFIN(&tpclda,&curq,POS(i),defbuf,deflen,deftype,NA,
                (ub2 *)0, (text *)0,NA,NA,(ub2 *)dlist[i]->rlen,(ub2
*)0);

  }
  return i;
}


/* process_select_list(): Fetch rows from a query.
*/
```

Appendix E

```
void process_select_list(num)
     int num;        /* number of select list items */
{

  int ntf;

  /* Print the headers for the query execution result */

  print_header(num);

  /* See if we need to limit the rows to fetch */

  ntf = (num_to_fetch >= 0) ? num_to_fetch : MAX_ARRAY;

  /* Fetch the rows and print them out */

  if ((ntf > MAX_ARRAY) || (num_to_fetch == -1)) {

    OEXFET(&tpclda,&curq,MAX_ARRAY,0,0);
    rows_ret = curq.rpc;
    print_rows(num,rows_ret);

    /* To avoid 1022 from OFEN */
    /* More rows to fetch...   */

    if (curq.rc != 1403) {
      if (num_to_fetch == -1) {
            while (OFEN(&tpclda,&curq, MAX_ARRAY) != -1) {
              print_rows(num,(curq.rpc-rows_ret));
              rows_ret = curq.rpc;
            }
            /* Print the final rows */
            print_rows(num,(curq.rpc-rows_ret));
            rows_ret = curq.rpc;
      } else {
            ntf -= MAX_ARRAY;
            while (OFEN(&tpclda,&curq, ((ntf>MAX_ARRAY) ?
MAX_ARRAY:ntf)) != -1) {
              ntf -= MAX_ARRAY;
              print_rows(num,(curq.rpc-rows_ret));
              rows_ret = curq.rpc;
              if (ntf <= 0) break;
            }
            print_rows(num,(curq.rpc-rows_ret));
            rows_ret = curq.rpc;
      }
    }
  } else {
    OEXFET(&tpclda,&curq,ntf,0,0);
    rows_ret = curq.rpc;
    print_rows(num,rows_ret);
  }

  fprintf(logfile,"\n\n%d row%c processed.\n", rows_ret,
            rows_ret == 1 ? '\0' : 's');

}


int OFEN(lda,cur,nrows)
     ldadef *lda;
     csrdef *cur;
     int     nrows;
{

  if(ofen(cur, nrows))
    if (cur->rc != 1403) {
      sql_error(lda,cur);
      return 1;
    } else
      return -1;

  return 1;
}


int get_statement()
{

  char line[128];
  char *pos, *str;

  /* Reset statement buffer */

  stmt[0] = '\0';

  while (fgets(line, 127, qtemp) != NULL) {

    /* skip blank lines */
    if (line[0] == '\n')
      continue;

    /* remove blanks */

    str = line;

    while (*str == ' ') str++;

    /* Let's get the line together first */

    strcat(stmt, str);

    /* if this is a comment line */
    if ((str[0] == '-') && (str[1] == '-'))
      return COMMENT;

    /* see if this is a set_fetchrows line */
    if (strncmp(str, "set_fetchrows", 13) == 0) {
      pos = strchr(str, ';');
      *pos = '\0';
      pos = strchr(str, '=');
      num_to_fetch = atol(++pos);
      return SET_FETCHROW;
    }

    /* if this is the end of the current statement */
    if ((pos = strchr(stmt, ';')) != NULL) {
      *pos = '\0';
      return SQL_STMT;
    }
  }
  return END_OF_FILE;
}
```

```
/* memalloc(): Allocates memory, exit program if we have a problem.
*/

void *memalloc(size)
     int size;
{

  void *tmp;

  if ((tmp = (void *) malloc(size)) == NULL) {
    fprintf(stderr, "Error in malloc\n");
    SQLexit();
    return NULL;       /* should never reach here */
  } else {
    return tmp;
  }
}



void print_header(nsel)
     int nsel;           /* Number of select list items */
{

  int i;
  int len = 0;           /* Running column length */
  int cwid = 0;

  fprintf(logfile, "\n");

  for (i=0; i<nsel; i++) {

    /* format the output a little */

    cwid = MAX(slist[i].dbsize, slist[i].buflen);

    /* do a little bit of formatting */

    if (cwid > 80) {
      fprintf(logfile,"\n");
      len = 0;
    } else if ((len += cwid) > 80) {
      fprintf(logfile,"\n");
      len = cwid;
    }
#ifdef FORMAT1
    if ((slist[i].dbtype == INT_TYPE) || (slist[i].dbtype ==
FLT_TYPE))
      fprintf(logfile, "%*s ", cwid, slist[i].buf);
    else  /* string type */
      fprintf(logfile, "%*s ", -cwid, slist[i].buf);
#else
      fprintf(logfile, "%*s ", -cwid, slist[i].buf);
#endif /* FORMAT1 */
  }

  fprintf(logfile,"\n");
}


void print_rows(ncol, nrow)
     int ncol;
     int nrow;
{

  int i,j;
  int len;
  int cwid;

  for (i=0;i<nrow;i++) {

    len = 0;

    for (j=0;j<ncol;j++) {

      cwid = MAX(slist[j].dbsize, slist[j].buflen);

      /* do a little bit of formatting */

      if (cwid > 80) {
            fprintf(logfile,"\n");
            len = 0;
      } else if ((len += cwid) > 80) {
            fprintf(logfile,"\n");
            len = cwid;
      }

      switch(slist[j].dbtype) {
      case INT_TYPE:
#ifdef HAVE_SCALE
            fprintf(logfile, "%*ld|", cwid,    (dlist[j]->ibuf)[i]);
            break;
#endif /* HAVE_SCALE */
      case FLT_TYPE:
#ifdef FORMAT1
            fprintf(logfile,"%*.2f ", cwid, (dlist[j]->fbuf)[i]);
#else
            fprintf(logfile,"%*.2f ", -cwid, (dlist[j]->fbuf)[i]);
#endif /* FORMAT1 */
            break;
      default:
            fprintf(logfile, "%*s ", -(cwid), (dlist[j]->sbuf)[i]);

            break;
      }
    }
    fprintf(logfile, "\n");
  }
}

/* remove_newline(): Remove newline character from str. */

void remove_newline(str)
     char *str;
{

  char *p;

  while ((p = strchr(str,'\n')) != NULL)
    *p = ' ';
}
```

qexecpl.h

```
/*==================================================================
+
|       Copyright (c) 1995  Oracle Corp, Redwood Shores, CA
|
|                    OPEN SYSTEMS PERFORMANCE GROUP
|
|                         All Rights Reserved
|

+==================================================================+
 | FILENAME
 |    qstream.h
 | DESCRIPTION
 |    SQL statement execution front-end header file.
 | MODIFIED
 |    pswong    03/07/96 - created

+==================================================================*
/

#ifndef QSTREAMPL_H

#define QSTREAMPL_H

#include <stdio.h>
#include <string.h>
/*#include <sys/param.h>*/
#include <sys/types.h>
#include <time.h>
#include <errno.h>
#include <math.h>

#include <oratypes.h>
#include <ocidfn.h>

#ifdef __STDC__
#include <ociapr.h>
#else
#include <ocikpr.h>
#endif /* __STDC__ */

typedef struct cda_def csrdef;
typedef struct cda_def ldadef;

/* some basic definitions */

#define UNAME_LEN 64
#define MAX_FILE_PATH_LEN 128

#ifndef TRUE
#define TRUE 1
#endif /* TRUE */

#ifndef FALSE
#define FALSE 1
#endif /* FALSE */

#define MAX(x,y) ((x >= y) ? x : y)
#define MIN(x,y) ((x <= y) ? x : y)

/* defines and typedefs for parsing */

#define CRT_TBL  1
#define INS_STMT 3
#define SEL_STMT 4
#define UPD_STMT 5
#define DRP_VIEW 7
#define DRP_TBL  8
#define DEL_STMT 9
#define CRT_VIEW 10

/* defines and typedefs for query description */

#define MAX_COLNAME_SIZE 16   /* Maximum length of Column name  */
#define MAX_SEL_LIST 16       /* Maximum items on a select list */

#define END_OF_LIST 1007      /* Error code when we reach the end of
the */
                              /* select list.
*/

/* types for describe */

#define CHAR_TYPE 1
#define NUM_TYPE  2
#define INT_TYPE  3
#define FLT_TYPE  4
#define STR_TYPE  5
#define DATE_TYPE 12

#define NUMWIDTH 16           /* Width of the numeric fields    */

#define POS(i) (i+1)          /* The position is 1...n instead  */
#define IND(i) (i-1)          /* of 0..n-1 as in an array.      */

typedef struct des
{
  sb4 dbsize;
  sb2 dbtype;
  sb1 buf[MAX_COLNAME_SIZE];
  sb4 buflen;
  sb2 dsize;
  sb2 precision;
  sb2 scale;
  sb2 nullok;
} sltype;


/* defines and typedefs for query select list definition */

#define MAX_ARRAY 3        /* Maximum array size for array fetch */

#define MAX_STR_LEN 256    /* Maximum size for string variables
*/
#define MAX_PREALLOC 8     /* Maximum number of preallocated select
list */
                           /* definitions.
*/

/* Note:  Well, I don't want to waste too much memory at the start.
*/
/*        Thus, I defined a MAX_PREALLOC so that we limit the amount
*/
/*        of memory used for the dlist array.
*/
/*        The program will dynamically allocate the remaining
members   */
/*        of the dlist array if necessary.
*/

/*        Actually, I can go one step further by making everything
*/
/*        dynamic.  But I don't want to call tons of mallocs and
frees  */
/*        during the query execution.  Another one of those memory-
CPU  */
/*        trade-offs.....
*/

#define INT sizeof(long)
#define STR sizeof(char)
#define FLT sizeof(double)

#define FLTP (double *)
#define INTP (long *)
#define STRP (char **)

typedef struct def
{
  long ibuf[MAX_ARRAY];
  double fbuf[MAX_ARRAY];
  char sbuf[MAX_ARRAY][MAX_STR_LEN];
  ub2 rlen[MAX_ARRAY];         /* return length     */
} dltype;

extern int errno;

#define SQL_LEN 2048

#ifndef NULL
#define NULL 0
#endif

#ifndef NULLP
# define NULLP (void *)NULL
#endif /* NULLP */

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#ifndef ub1
#define ub1 unsigned char
#endif

#define NA              -1      /* ANSI SQL NULL */
#define VER7            2
#define NOT_SERIALIZABLE  8177  /* ORA-08177: transaction not
serializable */

#define ADR(object) ((ub1 *)&(object))
#define SIZ(object) ((sword)sizeof(object))
#define SID(sid) ((sid == -1) ? 0 : sid)

/* For get_statement */

#define END_OF_FILE -1
#define COMMENT 1
#define SQL_STMT   2
#define SET_FETCHROW 3

#define OOPEN(lda,cursor)\
    if (oopen((cursor),(lda),(text*)0,NA,NA,(text*)0,NA))\
       {sql_error(lda,cursor);}\
    else\
       DISCARD 0

#define
ODEFIN(lda,cursor,pos,buf,bufl,ftype,scale,indp,fmt,fmtl,fmtt,rlen,r
code)\
    if (odefin((cursor),(pos),(ub1*)(buf),(bufl),(ftype),(scale),\
                    (sb2
*)(indp),(text*)(fmt),(fmtl),(fmtt),(rlen),(rcode)))\
       {sql_error(lda,cursor);}\
    else\
       DISCARD 0

#define OEXFET(lda,cursor,nrows,cancel,exact)\
    if (oexfet((cursor),(nrows),(cancel),(exact)))\
        {if ((cursor)->rc == 1403) DISCARD 0; \
         else {sql_error(lda,cursor);}} \
    else\
       DISCARD 0

#define OPARSE(lda,cursor,sqlstm,sql1,defflg,lngflg)\
    if (oparse((cursor),(text
*)(sqlstm),(sb4)(sql1),(defflg),(ub4)(lngflg)))\
       {sql_error(lda,cursor);}\
    else\
       DISCARD 0

#define OEXEC(lda,cursor)\
    if (oexec((cursor)))\
       {sql_error(lda,cursor);}\
    else\
       DISCARD 0

#define ISOTXT "alter session set isolation_level = serializable"

#endif /* QSTREAMPL_H */
```

gettime.c

```
/*================================================================+
 |       Copyright (c) 1996  Oracle Corp, Redwood Shores, CA      |
 |                    OPEN SYSTEMS PERFORMANCE GROUP              |
 |                         All Rights Reserved                    |
 +================================================================+
 | FILENAME
 |    gtime.c
 | DESCRIPTION
 |    Wrapper for gettime
 +================================================================*/

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

double gettime();

main() {
  printf("%f", gettime());
  exit(0);
}
```

uf1.pc

```
/*================================================================
+
 |       Copyright (c) 1995  Oracle Corp, Redwood Shores, CA
|
 |                    OPEN SYSTEMS PERFORMANCE GROUP
|
 |                         All Rights Reserved
|

+================================================================+
 | FILENAME
 |    uf1.pc
 | DESCRIPTION
 |    TPC-D benchmark uf1 driver, PRO*C version
 | MODIFIED
 |    pswong   03/25/96 - more polish
 |    pswong   02/28/96 - clean it up a little
 |    pswong   04/21/95           - update to Spec 9.1, Use QGEN


+================================================================*
/

#include <stdio.h>
#include <string.h>
#include <setjmp.h>
/*#include <sys/param.h>*/
#include <errno.h>
#include <math.h>
#include <string.h>
/*#include <unistd.h>*/
#include <signal.h>
#include <stdlib.h>

#include "shared.h"

#define MAX_FILE_PATH_LEN 256
#define LINESIZE 256
#define LSIZE (LINESIZE-1)
#define UNAME_LEN 32

/* Include the SQL Communications Area. */

#include <sqlca.h>

/* Inclde the Oracle Communication Area */

EXEC SQL INCLUDE oraca;
EXEC ORACLE OPTION (ORACA=YES);

extern double gettime();

/* Declare error handling functions */

void sql_error();
void usage();
int get_ord_line();
int get_item_line();
char *nextpos();
void wakeup();

#define MAX_ITEMS 40
#define MAX_VNAME_LEN 30
#define MAX_INAME_LEN 30

/* batch size for inserts */

#define MAX_ORD_INSERTS 100
#define MAX_LINE_INSERTS (7 * MAX_ORD_INSERTS)

#ifndef NULL
#define NULL 0
#endif

/* structures to store query results */
/* typedefs in query.h              */

EXEC SQL BEGIN DECLARE SECTION;
VARCHAR     lname[32];  /* UNAME_LEN username/passwd combo */

int linecnt = 0;   /* used to specify the number of rows for array
inserts */
int ordcnt = 0;    /* used to specify the number of rows for array
inserts */

/* declarations for ORDER */

long o_okey[100]; /*MAX_ORD_INSERTS];*/
long o_ckey[100]; /*MAX_ORD_INSERTS];*/
varchar o_ostat[100][2]; /*MAX_ORD_INSERTS][2];*/
float o_tprice[100]; /*MAX_ORD_INSERTS]; */
varchar o_odate[100][13]; /*MAX_ORD_INSERTS][DATE_LEN];*/
varchar o_opri[100][15+1]; /*MAX_ORD_INSERTS][O_OPRIO_LEN+1];*/
varchar o_clk[100][15+1]; /*MAX_ORD_INSERTS][O_CLRK_LEN+1];*/
int o_spri[100]; /*MAX_ORD_INSERTS]; */
varchar o_cmnt[100][79+1]; /*MAX_ORD_INSERTS][O_CMNT_MAX+1]; */

/* declarations for LINEITEM */

long l_okey[700]; /*MAX_LINE_INSERTS];*/
long l_pkey[700]; /*MAX_LINE_INSERTS];*/
long l_skey[700]; /*MAX_LINE_INSERTS];*/
int l_lnum[700]; /*MAX_LINE_INSERTS];*/
float l_quan[700]; /*MAX_LINE_INSERTS];*/
float l_eprice[700]; /*MAX_LINE_INSERTS];*/
float l_disc[700]; /*MAX_LINE_INSERTS];*/
float l_tax[700]; /*MAX_LINE_INSERTS];*/
varchar l_rflag[700][2]; /*MAX_LINE_INSERTS][2];*/
varchar l_lstat[700][2]; /*MAX_LINE_INSERTS][2];*/
varchar l_sdate[700][13]; /*MAX_LINE_INSERTS][DATE_LEN];*/
varchar l_cdate[700][13]; /*MAX_LINE_INSERTS][DATE_LEN];*/
varchar l_rdate[700][13]; /*MAX_LINE_INSERTS][DATE_LEN];*/
varchar l_sinst[700][25+1]; /*MAX_LINE_INSERTS][L_INST_LEN+1];*/
varchar l_smode[700][10+1]; /*MAX_LINE_INSERTS][L_SMODE_LEN+1];*/
varchar l_cmnt[700][44+1]; /*MAX_LINE_INSERTS][L_CMNT_MAX+1];*/

EXEC SQL END DECLARE SECTION;

int set_id;               /* set_id, global within the driver    */
int run_id;               /* run_id, global                      */
int proc_no;              /* process number, global              */
double sf = 1.0;          /* scale_factor, global                */
double tr_end = 0.0;      /* query end time                      */
double tr_start = 0.0;    /* query start time                    */
double product = 1.0;     /* cumulative product of query times   */
int recover = 0;          /* recover from previous UF2?          */

FILE *logfile;            /* log and report files                */
FILE *ordfile, *itemfile; /* input data files                    */

char itemline[LINESIZE];  /* temp storage for input LINEITEM row */

/* usage: prints the usage of the program */

void usage() {

  fprintf(stderr,"\nUsage: uf1.o[st]t <set_id> <run_id> <proc_no>
<scale factor> [f<path name for reports> u<uid/passwd>] \n\n");
  fprintf(stderr,"       set_id                 :the set id for this
update set\n");
  fprintf(stderr,"       run_id                 :the run id for this
TPCD run\n");
  fprintf(stderr,"       proc_no                :the process number
within this update stream\n");
  fprintf(stderr,"       scale factor           :scale factor for
the run\n\n");
  fprintf(stderr,"       r                      :indicates that this
is a recovery run\n");
  fprintf(stderr,"       f<path name for log>   :full path name for
reports\n\n");
  fprintf(stderr,"       u<uid/passwd>          :Username/Password
string - default is tcpd/tpcd\n\n");
  exit(-1);

}



void sql_error()
{

  /* ORACLE error handler */
  fprintf (stderr, "Error: SQL Error encountered in Oracle!\n");
  fprintf (stderr, "\n\n%.70s\n",sqlca.sqlerrm.sqlerrmc);
  fflush(stderr);

  EXEC SQL WHENEVER SQLERROR CONTINUE;
  EXEC SQL ROLLBACK WORK;
  exit(1);

}



void main(argc,argv)
int argc;
char *argv[];
{

  int orows = 0;
  int lrows = 0;
  int notdone = 1;
  int err = 0;
  char ordpath[MAX_FILE_PATH_LEN];
```

```
  char itempath[MAX_FILE_PATH_LEN];
  char logpath[MAX_FILE_PATH_LEN];

printf("begin uf1\n");
  /* Initialize some variables */

  strcpy((char *) lname.arr, "tpcd/tpcd");
  lname.len = strlen((char *)lname.arr);

  /* get the signal handler for the pause() */
  /* signal(SIGUSR1, wakeup);               */

  if ((argc > 8) || (argc < 5)) {
    usage();
  }

  /* argv[1] -- Set ID */

  if ((set_id = atoi(argv[1])) < 0 ) {
    usage();
    exit(-1);
  }

  /* argv[2] -- Run ID */

  run_id  = atoi(argv[2]);

  /* argv[3] -- Process Number */

  proc_no = atoi(argv[3]);

  /* argv[4] -- Scale Factor */

  sf = atof(argv[4]);

  /* Process  optional parameters */

  argc -= 4;
  argv += 4;

  while(--argc) {
    ++argv;
    switch(argv[0][0]) {
    case 'u':
      strncpy((char *) lname.arr, ++(argv[0]), UNAME_LEN);
      lname.len = strlen((char *)lname.arr);
      if (strchr((char *) lname.arr, '/') == NULL) {
            fprintf(stderr, "Login name must be in the format of
userid/passwd\n");
            usage();
            exit(-1);
      }
      break;
    case 'f':
      sprintf(logpath,"%s/tpcd%d.st%d.p%d.log", ++(argv[0]), run_id,
set_id, proc_no);
      break;
    case 'r':
      recover = 1;
      break;
    default:
      fprintf(stderr, "Unknown argument %s\n", argv[0]);
      usage();
      break;
    }
  }

  /* We cannot communicate back to the driver in a real run */

#ifdef DEBUG
  if ( (logfile = fopen(logpath,"a")) == NULL ) {
    fprintf(stderr,"Unable to open file '%s'\n", logpath);
    fprintf(stderr,"%s: %s\n", logpath, strerror(errno));
    exit(-1);
  }
#endif /* DEBUG */

  /* The input files by default should be located */
  /* in $TPCD/update/data.                */

  if (!recover) {
    sprintf(ordpath, "%s/order.tbl.u%d.%d",
              getenv("UPDATE_DIR"), set_id, proc_no);
    sprintf(itempath, "%s/lineitem.tbl.u%d.%d",
              getenv("UPDATE_DIR"), set_id, proc_no);
  } else {
    sprintf(ordpath, "%s/rorder.tbl.u%d.%d",
              getenv("UPDATE_DIR"), set_id, proc_no);
    sprintf(itempath, "%s/rlineitem.tbl.u%d.%d",
              getenv("UPDATE_DIR"), set_id, proc_no);
  }

  /* open those files */

  if ( (ordfile = fopen(ordpath,"r")) == NULL ) {
    fprintf(stderr,"Unable to open file '%s'\n", ordpath);
    fprintf(stderr,"%s: %s\n", ordpath, strerror(errno));
    exit(-1);
  }

  if ( (itemfile = fopen(itempath,"r")) == NULL ) {
    fprintf(stderr,"Unable to open file '%s'\n", itempath);
    fprintf(stderr,"%s: %s\n", itempath, strerror(errno));
    exit(-1);
  }

#ifdef DEBUG
  fprintf(logfile,"\t\tTiming Interval for Update Function 1\n");
  fprintf(logfile,"\t\t^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n\n");
  fprintf(logfile,"Query # \t Start time \t End time \t Elapsed
Time\n");
  fprintf(logfile,"------- \t ---------- \t -------- \t ------------
\n");
#endif /* DEBUG */

  /*  Establish sql_error() as the error handler. */

  EXEC SQL WHENEVER SQLERROR DO sql_error();

  /* Connect to ORACLE.  Program will call sql_error()
     if an error occurs in connecting to the default database. */
printf("attempting to connect to Oracle with %s\n",lname.arr);

  EXEC SQL CONNECT :lname;

#ifdef DEBUG
  printf("\nConnected to ORACLE as user: %s\n\n", lname.arr);
#endif /* DEBUG */

  /* To be added later on.                                      */
  /* The process now sits there and wait for a signal to start. */
  /* That signal is SIGUSR1.  The driver script should send it. */
  /* This is used as a sychronization mechanism.                */

  /* start doing the update */

#ifdef DEBUG
  tr_start = gettime();
#endif /* DEBUG */

  notdone = 1;   /* just in case */

  /* Need to get the first row from the lineitem input file first.
*/
  /* Yeah, I know it is not as elegant, but it works for now.
*/

  if (fgets(itemline, LSIZE, itemfile) == NULL) {
    fprintf(stderr, "Error: Unable to open LINEITEM input file
%s!\n",
                itempath);
    fflush(stderr);
    exit(-1);
  }

  while (notdone) {
    ordcnt = 0;
    linecnt = 0;

    /* Perform inserts to ORDERS first, but do not COMMIT until the
end */

    if ((ordcnt = get_ord_line()) < MAX_ORD_INSERTS) {
          notdone = 0;
      }

    /* Get all the rows from LINEITEM that corresponds to the set
*/
    /* of ORDERKEYs.
*/
    /* Perform inserts to LINEITEM, but do not COMMIT until the end
*/

    linecnt = get_item_line(ordcnt);

    if (ordcnt != 0) {
      err = 1;
      EXEC SQL FOR :ordcnt INSERT INTO orders VALUES
            (TO_DATE(:o_odate,'YYYY-MM-DD'), :o_okey, :o_ckey,
:o_opri,:o_spri,
          :o_clk, :o_ostat, :o_tprice, :o_cmnt);
    }
    orows += ordcnt;

    if (linecnt != 0) {
      err = 1;
      EXEC SQL FOR :linecnt INSERT INTO lineitem VALUES
            (TO_DATE(:l_sdate,'YYYY-MM-DD'), :l_okey, :l_disc,:
l_eprice,
          :l_skey, :l_quan, :l_rflag, :l_pkey, :l_lstat, :l_tax,
         TO_DATE(:l_cdate,'YYYY-MM-DD'), TO_DATE(:l_rdate,'YYYY-MM-
DD'),
          :l_smode, :l_lnum, :l_sinst,  :l_cmnt);
    }
    lrows += linecnt;

    /* Perform the Commit for this sub-transaction. */
    EXEC SQL COMMIT WORK;
  }

  /* end of this update function */

  if(recover) {
    printf("%d ORDERS row%c processed.\n", orows, orows == 1 ? '\0'
: 's');
    printf("%d LINEITEM row%c processed.\n", lrows, lrows == 1 ?
'\0' : 's');
  }

#ifdef DEBUG
  tr_end = gettime();
  printf("Ended Update Function 1 Process %d at %f\n", proc_no,
tr_end);
  printf("%d ORDERS row%c processed.\n", orows, orows == 1 ? '\0' :
's');
  printf("%d LINEITEM row%c processed.\n", lrows, lrows == 1 ? '\0'
: 's');
  fprintf(logfile,"\n\n%d ORDERS row%c processed.\n", orows,
            orows == 1 ? '\0' : 's');
  fprintf(logfile,"\n\n%d LINEITEM row%c processed.\n", lrows,
            lrows == 1 ? '\0' : 's');
  printf("Executed Update Function 1 Process %d in %f\n",
           proc_no, (tr_end - tr_start));
  fprintf(logfile,"%d\t\t%f\t%f\t  %f\n", proc_no,
            tr_start,tr_end,(tr_end - tr_start));
  fprintf(logfile,"\n-------------------------------------\n");
  fprintf(stdout,"\n-------------------------------------\n");
  fflush(logfile);
  fclose(logfile);
#endif /* DEBUG */

  EXEC SQL WHENEVER SQLERROR CONTINUE;

  /*  Disconnect from ORACLE. */

  EXEC SQL COMMIT WORK RELEASE;
  EXEC SQL WHENEVER SQLERROR DO sql_error();
  fclose(ordfile);
  fclose(itemfile);
  exit(0);
}


int get_ord_line () {

  int i;
  int rcnt = 0;
  char *pos1, *pos2;
  char line[LINESIZE];

  for (i=0; i<MAX_ORD_INSERTS; i++) {
    if (fgets(line, LSIZE, ordfile) == NULL) {
      return (i);
    }

    /* extract columns from the line                    */
```

```
    /* Can't use fscanf here because the delimiter is | */

    pos1 = nextpos(line);
    o_okey[i] = atol(line);

    pos2 = nextpos(pos1);
    o_ckey[i] = atol(pos1);
    pos1 = pos2;

    pos2 = nextpos(pos1);
    strcpy ((char *) o_ostat[i].arr, pos1);
    o_ostat[i].len = strlen((char *) o_ostat[i].arr);
    pos1 = pos2;

    pos2 = nextpos(pos1);
    o_tprice[i] = atof(pos1);
    pos1 = pos2;

    pos2 = nextpos(pos1);
    strcpy((char *) o_odate[i].arr, pos1);
    o_odate[i].len = strlen((char *) o_odate[i].arr);
    pos1 = pos2;

    pos2 = nextpos(pos1);
    strcpy((char *) o_opri[i].arr, pos1);
    o_opri[i].len = strlen((char *) o_opri[i].arr);
    pos1 = pos2;

    pos2 = nextpos(pos1);
    strcpy((char *) o_clk[i].arr, pos1);
    o_clk[i].len = strlen((char *) o_clk[i].arr);
    pos1 = pos2;

    pos2 = nextpos(pos1);
    o_spri[i] = atoi(pos1);
    pos1 = pos2;

    pos2 = nextpos(pos1);
    strcpy((char *) o_cmnt[i].arr, pos1);
    o_cmnt[i].len = strlen((char *) o_cmnt[i].arr);
    pos1 = pos2;
  }
  return (MAX_ORD_INSERTS);
}


int get_item_line (rowcnt)
     int rowcnt;
{

  int i=0, j;             /* at least one row will be scanned */
  char *pos1, *pos2;
  int notyet = 1;
  int okey;

  /* for each orderkey, extract the corresponding rows in LINEITEM
*/

  for (j=0; j<rowcnt; j++) {

    /* the first row should have been scanned here by now */
    /* either from the main program or from the previous  */
    /* invokation of this function                        */

    notyet = 1;

    while (notyet) {

      /* extract columns from the line */

      pos1 = nextpos(itemline);
      okey = atol(itemline);

      /* Compare the ORDERKEY values */

      if (okey == o_okey[j]) {

            l_okey[i] = okey;

            pos2 = nextpos(pos1);
            l_pkey[i] = atol(pos1);
            pos1 = pos2;

            pos2 = nextpos(pos1);
            l_skey[i] = atol(pos1);
            pos1 = pos2;

            pos2 = nextpos(pos1);
            l_lnum[i] = atoi(pos1);
            pos1 = pos2;

            pos2 = nextpos(pos1);
            l_quan[i] = atof(pos1);
            pos1 = pos2;

            pos2 = nextpos(pos1);
            l_eprice[i] = atof(pos1);
            pos1 = pos2;

            pos2 = nextpos(pos1);
            l_disc[i] = atof(pos1);
            pos1 = pos2;

            pos2 = nextpos(pos1);
            l_tax[i] = atof(pos1);
            pos1 = pos2;

            pos2 = nextpos(pos1);
            strcpy((char *) l_rflag[i].arr, pos1);
            l_rflag[i].len = strlen((char *) l_rflag[i].arr);
            pos1 = pos2;

            pos2 = nextpos(pos1);
            strcpy((char *) l_lstat[i].arr, pos1);
            l_lstat[i].len = strlen((char *) l_lstat[i].arr);
            pos1 = pos2;

            pos2 = nextpos(pos1);
            strcpy((char *) l_sdate[i].arr, pos1);
            l_sdate[i].len = strlen((char *) l_sdate[i].arr);
            pos1 = pos2;

            pos2 = nextpos(pos1);
            strcpy((char *) l_cdate[i].arr, pos1);
            l_cdate[i].len = strlen((char *) l_cdate[i].arr);
            pos1 = pos2;

            pos2 = nextpos(pos1);
            strcpy((char *) l_rdate[i].arr, pos1);
            l_rdate[i].len = strlen((char *) l_rdate[i].arr);
            pos1 = pos2;

            pos2 = nextpos(pos1);
            strcpy((char *) l_sinst[i].arr, pos1);
            l_sinst[i].len = strlen((char *) l_sinst[i].arr);
            pos1 = pos2;

            pos2 = nextpos(pos1);
            strcpy((char *) l_smode[i].arr, pos1);
            l_smode[i].len = strlen((char *) l_smode[i].arr);
            pos1 = pos2;

            pos2 = nextpos(pos1);
            strcpy((char *) l_cmnt[i].arr, pos1);
            l_cmnt[i].len = strlen((char *) l_cmnt[i].arr);
            pos1 = pos2;
      } else {

        /* reset so that we won't run into seg faults */

        itemline[strlen(itemline)] = '|';
            break;
      }

      /* increments array index */
      i++;

      /* get next line, if failed, return */
      if (fgets(itemline, LSIZE, itemfile) == NULL) {
            return(i);
      }
    }
  }

  return (i);
}

char *nextpos(start)
     char *start;
{
  char *mark = strchr(start,'|');

  *mark = '\0';
  mark++;

  return (mark);
}


void wakeup() {
  return;
}
```

uf2_del.pc

```
/*=================================================================
+
 |       Copyright (c) 1995  Oracle Corp, Redwood Shores, CA
|
 |                    OPEN SYSTEMS PERFORMANCE GROUP
|
 |                         All Rights Reserved
|

+================================================================+
 | FILENAME
 |    uf1_del.pc
 | DESCRIPTION
 |    TPC-D benchmark uf2 driver, PRO*C version
 |    Also used to restore uf1.
 | MODIFIED
 |    pswong    03/30/96 - polished version
 |    pswong    06/13/95 - version 2
 |    pswong           06/07/95 - created

+================================================================*
/

#include <stdio.h>
#include <string.h>
#include <setjmp.h>
/*#include <sys/param.h>*/
#include <errno.h>
#include <math.h>
#include <string.h>
#include <signal.h>
/*#include <unistd.h>*/
#include <stdlib.h>

#include "shared.h"

#define MAX_FILE_PATH_LEN 256
#define LINESIZE 256
#define LSIZE (LINESIZE-1)
#define UNAME_LEN 32

/* max length of SQL statement */

#define SQL_LEN 4096

/* Didn't use bind variables because Oracle will perform FTS with
the  */
/* DELETE statement as it cannot estimate the key range of the
BETWEEN */
/* statement.
*/

/* We need so many different statement types to avoid cases like
*/
/* L_ORDERKEY BETWEEN x AND x which was known to have problems.
*/
```

Appendix E

```
#define DELO_TXT1 "DELETE FROM ORDERS WHERE O_ORDERKEY BETWEEN %ld
AND %ld "
#define DELL_TXT1 "DELETE FROM LINEITEM WHERE L_ORDERKEY BETWEEN %ld
AND %ld "
#define DELO_TXT2 "DELETE FROM ORDERS WHERE O_ORDERKEY = %ld "
#define DELL_TXT2 "DELETE FROM LINEITEM WHERE L_ORDERKEY = %ld "
#define EQO_TXT "OR O_ORDERKEY = %ld "
#define EQL_TXT "OR L_ORDERKEY = %ld "
#define DEL_LEN 80   /* a little larger than it's necessary for
safety sake */
#define BTWO_TXT "OR O_ORDERKEY BETWEEN %ld AND %ld "
#define BTWL_TXT "OR L_ORDERKEY BETWEEN %ld AND %ld "
#define BTW_LEN 50   /* also a little large than necessary */

/* batch size for inserts
*/
/* This is the number of BETWEEN statements that can fit in the SQL
*/
/* statement.  The -1 in the end is just for safety.
*/

#define MAX_INS ((SQL_LEN - DEL_LEN) / BTW_LEN - 1)
#define MAX_INSERTS ((getenv("MAX_DELETE") == NULL) ? 2 :
((atoi(getenv("MAX_DELETE")) > MAX_INS) ? MAX_INS :
atoi(getenv("MAX_DELETE"))))

/* The frequency of commits.  Default is to commit every 4 times. */

#define     COMMIT_FREQ 4

/* Include the SQL Communications Area. */

#include <sqlca.h>

/* Inclde the Oracle Communication Area */

EXEC SQL INCLUDE oraca;
EXEC ORACLE OPTION (ORACA=YES);

extern double gettime();

/* Declare error handling functions */

void sql_error();
void usage();
int get_ord_line();
void wakeup();

#ifndef NULL
#define NULL 0
#endif

/* structures to store query results */
/* typedefs in query.h               */

EXEC SQL BEGIN DECLARE SECTION;

varchar     sqlstmt[4096]; /*SQL_LEN];*/     /* the delete statement
for orders */
varchar     sqlstmt2[4096]; /*SQL_LEN];*/    /* the delete statement
for lineitem */
VARCHAR     lname[32]; /*UNAME_LEN];*/   /* username/passwd combo */

int rowcnt = 0;    /* used to specify the number of rows for array
inserts */

EXEC SQL END DECLARE SECTION;

int set_id;               /* set_id, global within the driver    */
int run_id;               /* run_id, global                      */
int proc_no;              /* process number, global              */
int u_flag=2;             /* usage flag, global                  */
double sf = 1.0;          /* scale_factor, global                */

FILE *ordfile;            /* flatfile for ORDERS                 */


/* usage: prints the usage of the program */

void usage() {

  fprintf(stderr,"\nUsage: uf2[st]t <set_id> <run_id> <proc_no>
<scale factor> <usage flag> [u<uid/passwd>] \n\n");
  fprintf(stderr,"       set_id                 :the set id for this
update set\n");
  fprintf(stderr,"       run_id                 :the run id for this
TPCD run\n");
  fprintf(stderr,"       proc_no                :the process number
within this update stream\n");
  fprintf(stderr,"       scale factor           :scale factor for
the run\n\n");
  fprintf(stderr,"       usage flag             :enter 1 for uf2 and
2 for restoring uf1\n\n");
  fprintf(stderr,"       u<uid/passwd>          :Username/Password
string - default is tcpd/tpcd\n\n");
  exit(-1);
}


void sql_error()
{

  /* ORACLE error handler */
  fprintf (stderr, "Error: error encountered in Oracle!\n");
  fprintf (stderr, "\n\n%.70s\n",sqlca.sqlerrm.sqlerrmc);
  fflush (stderr);

  EXEC SQL WHENEVER SQLERROR CONTINUE;
  EXEC SQL ROLLBACK WORK;
  exit(1);

}



void main(argc,argv)
int argc;
char *argv[];
{

  int orows = 0;
  int lrows = 0;
  int o, l;
  int notdone = 1;
  int num_exec = 0;
  /* int err = 0; */
  char ordpath[MAX_FILE_PATH_LEN];

  /* Initialize some variables */
printf("Begin uf2\n");

  strcpy((char *) lname.arr, "tpcd/tpcd");
  lname.len = strlen((char *)lname.arr);

  /* signal(SIGUSR1, wakeup); */

  if ((argc > 7) || (argc < 6)) {
    usage();
  }

  /* argv[1] -- Set ID */

  if ((set_id  = atoi(argv[1])) < 0 ) {
    usage();
    exit(-1);
  }

  /* argv[2] -- Run ID */

  run_id  = atoi(argv[2]);

  /* argv[3] -- Process Number */

  proc_no = atoi(argv[3]);

  /* argv[4] -- Scale Factor */

  sf = atof(argv[4]);

  /* argv[5] -- Filename prefix for orderkeys    */
  /* The input file by default should be located */
  /* in $TPCD/update/data.                       */

  u_flag = atoi(argv[5]);

  switch(u_flag) {
  case 1:
    sprintf(ordpath, "%s/delete.%d.%d",
                getenv("UPDATE_DIR"), set_id, proc_no);
    break;
  case 2:
    sprintf(ordpath, "%s/okey.u%d.%d",
                getenv("UPDATE_DIR"), set_id, proc_no);
    break;
  default:
    fprintf(stderr, "Illegal usage flag!\n");
    exit(-1);
  }
printf("ordpath is %s\n",ordpath);

  /* Process  optional parameters */

  argc -= 5;
  argv += 5;

  while(--argc) {
    ++argv;
    switch(argv[0][0]) {
    case 'u':
      strncpy((char *) lname.arr, ++(argv[0]), UNAME_LEN);
      lname.len = strlen((char *)lname.arr);
      if (strchr((char *) lname.arr, '/') == NULL) {
           fprintf(stderr, "Login name must be in the format of
userid/passwd\n");
           usage();
           exit(-1);
      }
      break;
    default:
      fprintf(stderr, "Unknown argument %s\n", argv[0]);
      usage();
      break;
    }
  }

  /* open the files */

  if ( (ordfile = fopen(ordpath,"r")) == NULL ) {
    fprintf(stderr,"Unable to open file '%s'\n", ordpath);
    fprintf(stderr,"%s: %s\n", ordpath, strerror(errno));
    exit(-1);
  }

  /*  Establish sql_error() as the error handler. */

  EXEC SQL WHENEVER SQLERROR DO sql_error();

  /* Connect to ORACLE.  Program will call sql_error()
     if an error occurs in connecting to the default database. */

  EXEC SQL CONNECT :lname;

#ifdef DEBUG
  printf("\nConnected to ORACLE as user: %s\n\n", lname.arr);
#endif /* DEBUG */

  /* pause(); */

  /* start the restore from uf1 */

  /* delete from ORDERS, hold the commit */

  notdone = 1;   /* just in case */

  while (notdone) {
    rowcnt = 0;
    if ((rowcnt = get_ord_line()) < MAX_INSERTS) {
           notdone = 0;
      }

    sqlstmt.len = strlen((char *)sqlstmt.arr);
    sqlstmt2.len = strlen((char *)sqlstmt2.arr);

    /* Execute the deletes, hold commit */

    if (rowcnt != 0) {
printf("exec sql statement %s\n",sqlstmt.arr);
      EXEC SQL EXECUTE IMMEDIATE :sqlstmt;
      o = sqlca.sqlerrd[2];
printf("exec sql statement %s\n",sqlstmt2.arr);
      EXEC SQL EXECUTE IMMEDIATE :sqlstmt2;
      l = sqlca.sqlerrd[2];
    }

    /* Commit once in a while */
```

```
    if (!((++num_exec)%COMMIT_FREQ)) {
      EXEC SQL COMMIT WORK;
    }

    orows += o;
    lrows += l;

  }

  /* COMMIT all deletes */

  EXEC SQL COMMIT WORK;
  EXEC SQL WHENEVER SQLERROR DO sql_error();

  /* Print the number of rows processed when we are recovering */

  if (u_flag == 2) {
    fprintf(stdout, "%d row%s deleted from ORDERS\n", orows,
              (orows == 1 ? "\0" : "s"));
    fprintf(stdout, "%d row%s deleted from LINEITEM\n", lrows,
              (orows == 1 ? "\0" : "s"));
  }

#ifdef DEBUG
  fprintf(stdout, "%d row%s deleted from ORDERS\n", orows,
            (orows == 1 ? "\0" : "s"));
  fprintf(stdout, "%d row%s deleted from LINEITEM\n", lrows,
            (orows == 1 ? "\0" : "s"));
#endif /* DEBUG */
printf("%d row%s deleted from ORDERS\n",orows,(orows == 1 ? "\0" :
"s"));
printf("%d row%s deleted from LINEITEM\n",lrows,(orows == 1 ? "\0" :
"s"));

  /* end of the restore */

  EXEC SQL WHENEVER SQLERROR CONTINUE;

  /*  Disconnect from ORACLE. */

  EXEC SQL COMMIT WORK RELEASE;
  EXEC SQL WHENEVER SQLERROR DO sql_error();
  fclose(ordfile);
  exit(0);
}


int get_ord_line (void) {

  int i;
  int rcnt = 0;
  long ord1, ord2;
  char *pos1;
  char line[LINESIZE];
  char o_stmt[BTW_LEN];
  char l_stmt[BTW_LEN];

printf("begin get_ord_line\n");
fflush(stdout);
  for (i=0; i<MAX_INSERTS; i++) {
    if (fgets(line, LSIZE, ordfile) == NULL) {
      return (i);
    }

printf("line is %s\n",line);
fflush(stdout);
    /* extract columns from the line */

    pos1 = strchr((char *) line, '|');
    *pos1 = '\0';
    pos1++;
    ord1 = atol(line);
    *(strchr((char *) pos1, '|')) = '\0';
    ord2 = atol(pos1);

printf("i = %d i is either 0 or not\n",i);
fflush(stdout);
    if (i == 0) {
      /* prepare statement for ORDERS and LINEITEM */
printf("ord1 is %d ord2 is %d\n",ord1,ord2);
fflush(stdout);
      if (ord1 == ord2) {
           sprintf((char *) sqlstmt.arr, DELO_TXT2, ord1);
           sprintf((char *) sqlstmt2.arr, DELL_TXT2, ord1);
      } else {
           sprintf((char *) sqlstmt.arr, DELO_TXT1, ord1, ord2);
           sprintf((char *) sqlstmt2.arr, DELL_TXT1, ord1, ord2);
      }
    } else {
      /* add next BETWEEN statement for ORDERS and LINEITEM */
printf("ord1 is %d ord2 is %d\n",ord1,ord2);
fflush(stdout);
      if (ord1 == ord2) {
           sprintf(o_stmt, EQO_TXT, ord1, ord2);
           strcat((char *) sqlstmt.arr, o_stmt);
           sprintf(l_stmt, EQL_TXT, ord1, ord2);
           strcat((char *) sqlstmt2.arr, l_stmt);
      } else {
           sprintf(o_stmt, BTWO_TXT, ord1, ord2);
           strcat((char *) sqlstmt.arr, o_stmt);
           sprintf(l_stmt, BTWL_TXT, ord1, ord2);
           strcat((char *) sqlstmt2.arr, l_stmt);
      }
    }

  }
printf("end get_ord_line\n");
fflush(stdout);
  return (MAX_INSERTS);
}


void wakeup() {
   return;
}
```

runuf1.sh

```
#!/bin/sh
#
#=====================================================================+
#        Copyright (c) 1995  Oracle Corp, Redwood Shores, CA          |
#                      OPEN SYSTEM PERFORMANCE GROUP                  |
#                           All Rights Reserved                       |
#=====================================================================+
# FILENAME
#    runuf1.sh
# DESCRIPTION
#    runuf1.sh -l [<path name for reports>] -u [<uid/passwd>]
#              -p [<program>] <run_id> <scale factor> <pair number>
#              <parallelism>
# USAGE
#    To execute UF1.
#=====================================================================

O=$ORACLE_HOME
TPCD_DIR=${TPCD_UPDATE_DATA}/tpcd

SCRIPT_DIR=${TPCD_DIR}/update/scripts
SRC_DIR=${TPCD_DIR}/update/source
GTIME_DIR=${TPCD_DIR}/source
GTIME=${GTIME_DIR}/gtime

export UPDATE_DIR

usage() {
echo ""
echo "runuf1.sh -l [<path name for reports>] -u [<uid/passwd>]"
echo "          -p [<program>] -h <run_id> <scale factor> <pair
number>"
echo "          <parallelism>"
echo ""
echo "run_id       : Run_ID of this update run."
echo "scale factor : Scale Factor of the database."
echo "set number   : The update pair number that this update
function belongs to."
echo "parallelism  : The parallelism of the updates."
echo ""
echo "-p           : Program to execute. Default is
$ORACLE_HOME/tpcd/update/source/uf1"
echo "-l           : Path name for reports on the update function.
Debug use for UF1 only"
echo "-u           : Userid/Password for Oracle. Default is
tpcd/tpcd."
echo "-h           : To Display this message."
echo ""
}


PROG=${SRC_DIR}/uf1
LOGPATH=.
PASSWD="tpcd/tpcd"

set -- `getopt "l:p:u:h" "$@"` || usage

while :
do
   case "$1" in
   -u) shift; PASSWD=$1;;
   -l) shift; LOGPATH=$1;;
   -p) shift; PROG=$1;;
   -h) usage; exit 0;;
   --) shift; break;;
   esac
   shift;
done

if [ $# -lt 4 ]
then
   usage
   exit 1
fi

RUN_ID=$1
SF=$2
SETNUM=$3
PAR=$4

if [ ${SF} -eq 0 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/tenthgig
fi

if [ ${SF} -eq 1 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/onegig
fi

if [ ${SF} -eq 2 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/twogig
fi

if [ ${SF} -eq 3 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/threegig
fi

if [ ${SF} -eq 10 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/tengig
fi

if [ ${SF} -eq 30 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/thirtygig
fi

if [ ${SF} -eq 100 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/hundgig
fi

if [ ${SF} -eq 300 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/threehundgig
fi

if [ ${SF} -eq 1000 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/onetbyte
fi
```

```
i=1
PID=""

# perform the update function 1

START=`$GTIME`

while [ $i -le $PAR ]
do

# Kick off the program first, but wait for the signal to start.
# In an OPS environment, we may want to rsh the PROG to the
# different nodes.

   ${PROG} ${SETNUM} ${RUN_ID} ${i} ${SF} f${LOGPATH} u${PASSWD} &
   PID="$PID $!"
   i=`expr $i + 1`

done

# now wait for start signal (SIGKILL will do)

# ./trig

# If we haven't timed out, start everyone

#if [ $? -ne 1 ]
#then
#   kill -USR1 $PID
#else
#   echo ""
#   echo "Update Function 1 Startup failed!"
#   echo "Failed to receive start signal"
#   echo ""
#fi

# Program has started, now wait for all the update processes to
finish.

wait

END=`$GTIME`

# Done

echo ""
echo "Update Function 1 Set $SETNUM done!"
echo "Elapsed Time is `echo $END - $START | bc`"
echo ""
```

runuf2.sh

```
#!/bin/sh
#
#==================================================================+
#        Copyright (c) 1995  Oracle Corp, Redwood Shores, CA       |
#                    OPEN SYSTEM PERFORMANCE GROUP                 |
#                         All Rights Reserved                      |
#==================================================================+
# FILENAME
#    runuf2.sh
# DESCRIPTION
#    runuf2.sh [-u <uid/passwd to login>] [-p <program>] <run_id>
#              <scale factor> <pair number> <parallelism>
# USAGE
#    To execute UF2.
#==================================================================
O=$ORACLE_HOME
TPCD_DIR=${TPCD_UPDATE_DATA}/tpcd

SCRIPT_DIR=${TPCD_DIR}/update/scripts
SRC_DIR=${TPCD_DIR}/source
GTIME_DIR=${TPCD_DIR}/source
GTIME=${GTIME_DIR}/gtime

export UPDATE_DIR

usage() {
echo ""
echo "runuf2.sh [-u <user/passwd>] [-p <program>] <run_id> <scale
factor> <pair number> <parallelism>"
echo ""
echo "run_id       : Run_ID of this update run."
echo "scale factor : Scale Factor of the database."
echo "set number   : The update pair number that this update
function belongs to."
echo "parallelism  : The parallelism of the updates."
echo ""
echo "-p           : Program to execute. Default is
$ORACLE_HOME/tpcd/update/source/uf2_del."
echo "-u           : Userid/Password for Oracle. Default is
tpcd/tpcd"
echo "-h           : To Display this message."
echo ""
}

PROG=${SRC_DIR}/uf2
PASSWD="tpcd/tpcd"

set -- `getopt "p:u:h" "$@"` || usage

while :
do
   case "$1" in
   -u) shift; PASSWD=$1;;
   -p) shift; PROG=$1;;
   -h) usage; exit 0;;
   --) shift; break;;
   esac
   shift;
done

if [ $# -lt 4 ]
then
   usage
   exit 1
fi

RUN_ID=$1
SF=$2
SETNUM=$3
PAR=$4

if [ ${SF} -eq 0 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/tenthgig
fi

if [ ${SF} -eq 1 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/onegig
fi

if [ ${SF} -eq 2 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/twogig
fi

if [ ${SF} -eq 3 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/threegig
fi

if [ ${SF} -eq 10 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/tengig
fi

if [ ${SF} -eq 30 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/thirtygig
fi

if [ ${SF} -eq 100 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/hundgig
fi

if [ ${SF} -eq 300 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/threehundgig
fi

if [ ${SF} -eq 1000 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/onetbyte
fi

i=1
PID=""

START=`$GTIME`

# perform the update function

while [ $i -le $PAR ]
do

# Kick off the program first, but wait for the signal to start

   ${PROG} ${SETNUM} ${RUN_ID} ${i} ${SF} 1 u${PASSWD} &
   PID="$PID $!"
   i=`expr $i + 1`

done

# now wait for start signal (SIGKILL will do)

#./trig

# If we haven't timed out, start everyone

#if [ $? -ne 1 ]
#then
#   kill -USR1 $PID
#else
#   echo ""
#   echo "Update Function 2 Startup failed!"
#   echo "Failed to receive start signal"
#   echo ""
#fi

# Program has started, now wait for all the update processes to
finish.

wait

END=`$GTIME`

# Done

echo ""
echo "Update Function 2 Set $SETNUM done!"
echo "Elapsed Time is `echo $END - $START | bc`"
echo ""
```

runpower1.sh

```
#!/bin/sh
#
#================================================================+
#         Copyright (c) 1996  Oracle Corp, Redwood Shores, CA    |
#                    OPEN SYSTEMS PERFORMANCE DIVISION           |
#                         All Rights Reserved                    |
#                                                                |
#================================================================+
#  FILENAME
#     runpower1.sh
#  DESCRIPTION
#     Usage: runpower1.sh [-p <program for query stream>]
#                         [-u1 <program for UF1>] [-u2 <program for
UF2>]
#                         [-o] [-s] [-h] [-u <user/password>]
#                            <scale factor> <update parallelism>
#
#     Single stream TPC-D Power Test.
#
#================================================================+

ORACLE_HOME=d:/orant
TPCD_HOME=/tpcd
SCRIPT_DIR=${TPCD_HOME}/scripts
SQL_DIR=${TPCD_HOME}/sql
UPD_DIR=${TPCD_HOME}/update
SRC_DIR=${TPCD_HOME}/source

RUN_ID_FILE=${TPCD_HOME}/r_id

UPD_SQL=${UPD_DIR}/sql
UPD_SPT=${TPCD_HOME}/update/scripts
UPD_SRC=${TPCD_HOME}/update/source
UPD_DAT=${UPD_DIR}/data

TPCD_BIN=${TPCD_HOME}/bin
TPCD_LOG=${TPCD_HOME}/log
TPCD_RPT=${TPCD_HOME}/rpt

OUT=${TPCD_HOME}/out

GTIME=${SRC_DIR}/gtime

DF=/dev/null
HID=1
INTERVAL=60
COUNT=1200

# The defaults

USER="tpcd/tpcd"
QPROG=${SRC_DIR}/qexec.exe

U1PROG=${UPD_SRC}/uf1.exe
U2PROG=${UPD_SRC}/uf2.exe

usage () {

echo " "
echo "Usage: $0 [-p <program for query stream>] [-u1 <program for
UF1>]"
echo "          [-u2 <program for UF2>] [-o] [-s] [-h] [-u
<user/password>]"
echo "             <scale factor> <update parallelism>"
echo ""
echo "scale factor     : The scale factor of the run."
echo "update ||ism     : The parallelism to use for the UFs."
echo ""
echo "-p <program>     : Program for Query Stream."
echo "                   Default is $QPROG."
echo "-u1 <program>    : Program for UF1."
echo "                   Default is $U1PROG."
echo "-u2 <program>    : Program for UF2."
echo "                   Default is $U2PROG."
echo "-o               : Collect Oracle statistics."
echo "-s               : Collect System statistics."
echo "-u <user/passwd> : User/Password. Default is tpcd/tpcd."
echo "-h               : Displays this message."
}
set -- `getopt "p:u1:u2:u:osh" "$@"` || usage

while :
do
   case "$1" in
   -u1) shift; U1PROG=$1;;
   -u2) shift; U2PROG=$1;;
   -p) shift; QPROG=$1;;
   -o) shift; OSTAT=1;;
   -s) shift; SSTAT=1;;
   -h) usage; exit 0;;
   --) shift; break;;
   esac
#shift;
done

if [ "$#" -ne "2" ]
then
  usage
  exit 1
fi

SF=$1
PARA=$2
THROUGHPUT=1

if [ $SF -eq 1 ]
  then
QRY_DIR=${TPCD_HOME}/queries/q_one_0.sql
INIT_DIR=${TPCD_HOME}/dbs/onegig
elif [ $SF -eq 2 ]
  then
QRY_DIR=${TPCD_HOME}/queries/q_two_0.sql
INIT_DIR=${TPCD_HOME}/dbs/twogig
elif [ $SF -eq 3 ]
  then
QRY_DIR=${TPCD_HOME}/queries/q_three_0.sql
INIT_DIR=${TPCD_HOME}/dbs/three
elif [ $SF -eq 10 ]
  then
QRY_DIR=${TPCD_HOME}/queries/q_ten_0.sql
INIT_DIR=${TPCD_HOME}/dbs/tengig
elif [ $SF -eq 100 ]
  then
QRY_DIR=${TPCD_HOME}/queries/q_hund_0.sql
INIT_DIR=${TPCD_HOME}/dbs/hundgig
elif [ $SF -eq 300 ]
  then
QRY_DIR=${TPCD_HOME}/queries/q_threehund_0.sql
INIT_DIR=${TPCD_HOME}/dbs/threehund
elif [ $SF -eq 1000 ]
  then
QRY_DIR=${TPCD_HOME}/queries/q_tbyte_0.sql
INIT_DIR=${TPCD_HOME}/dbs/tbyte
fi

export SF PARA THROUGHPUT

#${UPD_SPT}/genuf1.sh 1 ${PARA} ${SF}
#${UPD_SPT}/genuf1.sh -r 1 ${PARA} ${SF}
#${UPD_SPT}/genuf2.sh 1 ${PARA} ${SF}

if [ ! -f $RUN_ID_FILE ]
then
   echo "0" > $RUN_ID_FILE
fi

RUN_ID=`cat $RUN_ID_FILE`
RUN_ID=`expr $RUN_ID + 1`
echo $RUN_ID > $RUN_ID_FILE

echo "TPC-D Power Test Run `date`"
echo "RUNID is $RUN_ID"
echo ""

if [ $OSTAT ]
then
    svrmgr23 @${ORACLE_HOME}/rdbms73/admin/utlbstat
fi

if [ $SSTAT ]
then
    ${TPCD_BIN}/start_ntstat.sh $RUN_ID
fi

START=`$GTIME`
echo "TPC-D Power Test Execution starts at $START"
echo ""

# Execute UF1

echo "Start UF1 at `date`"

sh ${UPD_SPT}/runuf1.sh -p ${U1PROG} -u ${USER} ${RUN_ID} ${SF} 1
${PARA} \
            > ${OUT}/uf1.${RUN_ID}.${HID} 2>&1

# Execute Query Stream

echo "End UF1. Start Query Stream at `date`"

${QPROG} ${USER} q${QRY_DIR} r${TPCD_RPT}/rpt.${RUN_ID}.${HID} \
          1${OUT}/qs.${RUN_ID}.${HID} 2>&1

# Execute UF2

echo "End Query Stream. Start UF2 at `date`"

sh ${UPD_SPT}/runuf2.sh -p ${U2PROG} -u ${USER} ${RUN_ID} ${SF} 1
${PARA} \
            > ${OUT}/uf2.${RUN_ID}.${HID} 2>&1

echo "END UF2 `date`"

END=`$GTIME`

echo "TPC-D Power Test Execution ends at $END"
echo "Measurement Interval is `echo $END - $START | bc`"
echo ""
echo "-------------------------------------------------"
echo ""

if [ $OSTAT ]
then
    svrmgr23 @${ORACLE_HOME}/rdbms73/admin/utlestat
    mv report.txt ora_stat_${RUN_ID}.txt
fi

if [ $SSTAT ]
then
    ${TPCD_BIN}/kill_ntstat.sh ${RUN_ID}
    mv report_os.txt sys_stat_${RUN_ID}.txt
fi

echo "-- update function 1" >> ${TPCD_RPT}/rpt.${RUN_ID}.${HID}
awk -f utime.awk ${OUT}/uf1.${RUN_ID}.${HID} >>
${TPCD_RPT}/rpt.${RUN_ID}.${HID}
echo "-- update function 2" >> ${TPCD_RPT}/rpt.${RUN_ID}.${HID}
awk -f utime.awk ${OUT}/uf2.${RUN_ID}.${HID} >>
${TPCD_RPT}/rpt.${RUN_ID}.${HID}
cat ${TPCD_RPT}/rpt.${RUN_ID}.${HID} >
${TPCD_RPT}/pt_rpt.${RUN_ID}.${HID}
${TPCD_BIN}/metric ${SF} < ${TPCD_RPT}/rpt.${RUN_ID}.${HID} >>
${TPCD_RPT}/pt_rpt.${RUN_ID}.${HID}

rm -f ${TPCD_RPT}/rpt.${RUN_ID}.${HID}
cat ${INIT_DIR}/p_run.ora >> ${TPCD_RPT}/pt_rpt.${RUN_ID}.${HID}
```

Appendix F: Initial Ten Rows

```
O_ORDERDA O_ORDERKEY O_CUSTKEY  O_ORDERPRIORITY O_SHIPPRIO O_CLERK         O O_TOTALPRI O_COMMENT
--------- ---------- ---------- ---------------- ---------- --------------- - ---------- ----------------------------------------------------------------------------
14-APR-98  360000001   12141070 5-LOW                     0 Clerk#000096787 O  137468.72 4S2iL5jR5jLjxzNN3Pw6y1LkP2LhP2gLi6PAzjn
17-AUG-92  360000002    5751661 2-HIGH                    0 Clerk#000091388 F  189568.04 xR120jh6l5lxh2wO4POly3
29-DEC-97  360000003    8155192 1-URGENT                  0 Clerk#000044232 O  158220.58 zl1ORL47NxwwA4PAg7C5kQQLLxxhR22xx110gixClOP5 O676z50RkiO g71PLNCNz1R
27-OCT-93  360000004    6783572 2-HIGH                    0 Clerk#000003362 F  104373.18 iNL73lgykwPMxgM1Almz0
Ligjn2A7AS25LMxm2ml5Pmhzn00RnhQMwxSg14CQB2R3QNN7mL4Nwwm
12-MAR-98  360000005   12598655 2-HIGH                    0 Clerk#000003276 O  221133.89 Ay402MOiwg6LmzQmCy4g2l6m5jgLLB4N602wM
07-MAY-95  360000006    5587685 4-NOT SPECIFIED           0 Clerk#000057277 P  277913.51 gmmmxSRSk5PQiMhhkwS07Ok16yMQ026mR2g Cy5kSMnnLhP4mR7hM2
09-AUG-93  360000007   12213914 4-NOT SPECIFIED           0 Clerk#000048694 F   54621.5 zkPCQBB yMCxxCOy 6zxOjLwn1ji7j5gOB3h7lRCO1N
08-OCT-96  360000032    4238543 1-URGENT                  0 Clerk#000084925 O  273499.96 6x57C3A0gn7gw0Ah gBki05mxN Rh6m
12-MAY-96  360000033    4983988 5-LOW                     0 Clerk#000032416 O   48913.21 SNNBh3kniLAMkRSwzknx4NQmz66yBOm5 340mzA NmRhnw2B6M53ikgnx50OkMA2hz7yy
17-MAY-92  360000034    5874328 1-URGENT                  0 Clerk#000010909 F   55271.8 C6QNhhl6AkjShi7m innBSlNN1R ig6 x1zwL1Rlki70n2zmMjnSPL5zB1hC1P6l1x4ghiBl
10 rows selected.
L_SHIPDAT L_ORDERKEY L_DISCOUNT L_EXTENDED L_SUPPKEY  L_QUANTITY L L_PARTKEY  L L_TAX      L_COMMITD L_RECEIPT L_SHIPMODE L_LINENUMB L_SHIPINSTRUCT
L_COMMENT
--------- ---------- ---------- ---------- ---------- ---------- - ---------- - --------- --------- --------- ---------- ---------- ------------------------- --
------------------------------------------
23-SEP-97  165000001        .05   22230.05     874399         17 N   14624384 O        .08 17-SEP-97 03-OCT-97 FOB                 1 NONE
5nQNNxnS7hx llwxiixh
31-AUG-97  165000001        .08    5843.76     518568          4 N   12018543 O        .08 04-SEP-97 29-SEP-97 RAIL                2 COLLECT COD
j6z Q3kxNx
22-JUL-97  165000001        .04   25196.73     396039         13 N   15645993 O        .05 31-AUG-97 18-AUG-97 TRUCK               3 COLLECT COD
m66i 0 Rw4QOMBw1lPOkQ2
02-AUG-97  165000001        .02   23498.93     447713         13 N    2197710 O        .03 17-AUG-97 26-AUG-97 TRUCK               4 NONE
kCk07QxM6B3wmhQgj7NA5jzPL5 7yLMBBhgQwAMRnL
24-JUL-97  165000001          0   26439.26     657434         19 N   16907385 O        .05 12-SEP-97 29-JUL-97 REG AIR             5 NONE
3ARi4AR4kwzA7Og3N0jBmSPyS1m 1g4R
02-SEP-97  165000001         .1   66460.68     418593         44 N    2418592 O        .04 23-AUG-97 26-SEP-97 TRUCK               6 DELIVER IN PERSON
w2CRRLOmnP6k30n0h
26-JUL-97  165000001        .05    16462.2     512694         10 N    9262684 O        .03 15-AUG-97 30-JUL-97 SHIP                7 DELIVER IN PERSON
zPln0C2Cg7njkMOm k
21-SEP-96  165000002        .03   86771.65     423783         49 N   17923748 O        .03 06-AUG-96 26-SEP-96 AIR                 1 NONE
R66kRCCxLn65QA0PAQz
09-AUG-96  165000002        .07   50841.36     429484         36 N    4429483 O        .06 13-SEP-96 07-SEP-96 REG AIR             2 COLLECT COD
72lzijn0RhBki6m R73j1k03
09-JUL-96  165000002         .1   29950.08      60257         24 N    6310238 O        .04 14-SEP-96 01-AUG-96 AIR                 3 TAKE BACK RETURN
Qwgk6Q7jPMm75SN241wm5mn
10 rows selected.
C_CUSTKEY  C_MKTSEGME C_NATIONKE C_NAME                   C_ADDRESS                                 C_PHONE         C_ACCTBAL  C_COMMENT
---------- ---------- ---------- ------------------------ ----------------------------------------- --------------- ---------- -------------------------------
------------------------------------------
  12000001 FURNITURE          10 Customer#012000001       QlMklCOASiBCO                             20-467-775-1131    6012.48
mAP6j125xN4ONwMhSkl02nLNgON7NAiO2iB6SkR4zliB g60ln4AQA471 CNO6lzz164MzO35y
  12000002 MACHINERY           4 Customer#012000002       Nn0PNRQ56 7LkwOL                          14-883-132-9248    3722.74
ghLOwkhMQ4AmgjAmlSzzlRQmj3B2700lyC6iQzQ
  12000003 BUILDING            2 Customer#012000003       2z4k4L PmzNOzmnmPmzOSL1                   12-818-591-4368    9085.06 7LR7gB iOzjSiOyOC
jA10M3gPPMAixw10SmkP C zRSy ky47R1N5h6PL1mO2y7SkRyNO0nLRmChPzj
  12000004 BUILDING            5 Customer#012000004       NCCBlNPRzxkQRi7Silj2nORlj                 15-765-367-1664    -273.06 n2
iznLh65CP161zmOQzz4gxRLxAhCM3lCw114 Ag 6jx2n5Nkxy4z
  12000005 MACHINERY           7 Customer#012000005       SwR3BjmBwQnixzk0SM                        17-910-236-8625    6607.49 36361 i0nCRylzzyi7C4
ARL1wmzkiO41QwzNNACzggQA1Cj6iAh
  12000006 BUILDING            0 Customer#012000006       1 Q1gCL3h2NzgyNxk yCm47Lm5x4g 7xlOLO2     10-744-327-4219    5162.13 yCQzzwhzzBLCmMyjCLg
PySO2i02gL6kLz77NRQzLn3jyMNC7ii1 jPm3SLyRiOmB BgA3SOM3BR026n
  12000007 FURNITURE          12 Customer#012000007       1AS3imgmLA71O                             22-512-408-3565     959.53 wCl57 M1RwilN 1NxARz 1wM0mgzBw Q2l
m5ghzljR wm Awx 01n
  12000008 MACHINERY           1 Customer#012000008       574j6Pwyy6370zBnAiNMCixzBS                11-102-340-3802    9710.97 yNB3ilyQP1yRwni7QO1R4NQRynB0S
3kSRmi54hi xBSzN0gm
  12000009 FURNITURE          11 Customer#012000009       Ag0zm4wL5gB11ML0niBO2Snn11khOL33L         21-884-830-4798    3407.17
jNiM6yOiPC7PhRlO32Q1y6MC5wzw7O4gxzhizSwONCny2ONjSlPl551B3L
  12000010 FURNITURE          20 Customer#012000010       22ji4Ay7MMB7wwCOhxi0nniSgC0wgnjLlOORMmmC 30-678-745-9724    7350.12  5kLOQ4xz1nS6 6351P4QyzP5Q1Bn50ij
CL5NL6yPl
10 rows selected.
P_PARTKEY  P_TYPE                    P_SIZE     P_BRAND    P_NAME                                                P_CONTAINE P_MFGR
P_RETAILPR P_COMMENT
---------- ------------------------- ---------- ---------- ----------------------------------------------------- ---------- ------------------------- ---------
- ----------------------
   6000001 LARGE BRUSHED COPPER              20 Brand#55   aquamarine saddle khaki chartreuse purple             JUMBO PKG  Manufacturer#5
1100.71 3lxlOhQ
   6000002 STANDARD POLISHED STEEL           48 Brand#52   maroon seashell smoke dim papaya                      LG BOX     Manufacturer#5
1101.71 4yRCMxCwmMnN1g67P3NO
   6000003 PROMO BURNISHED NICKEL            20 Brand#11   navy chiffon frosted orchid                           WRAP JAR   Manufacturer#1
1102.71 w57Ox6m5S27P
   6000004 LARGE PLATED COPPER               37 Brand#14   azure lace burlywood firebrick rosy                   LG PKG     Manufacturer#1
1103.71 wQlBL
   6000005 MEDIUM POLISHED BRASS             37 Brand#14   burnished wheat metalic chocolate blush               SM BOX     Manufacturer#1
1104.71 LQki2L
   6000006 ECONOMY PLATED NICKEL             13 Brand#35   smoke dodger steel royal                              LG CASE    Manufacturer#3
1105.71 mww306B32Njim
   6000007 MEDIUM POLISHED TIN               33 Brand#33   tan blush peach cyan                                  LG JAR     Manufacturer#3
1106.71 SRnS6klj
   6000008 PROMO POLISHED NICKEL             37 Brand#45   honeydew midnight blue white                          SM PKG     Manufacturer#4
1107.71 CLB255
   6000009 MEDIUM ANODIZED TIN               39 Brand#12   misty seashell salmon violet orchid                   JUMBO BAG  Manufacturer#1
1108.71 jBAPyA52OLy
   6000010 ECONOMY BURNISHED BRASS            2 Brand#34   black seashell violet lavender tomato                 SM BOX     Manufacturer#3
1109.72 gwgkNzCz
10 rows selected.
S_SUPPKEY  S_NATIONKE S_COMMENT                                                                                 S_NAME                    S_ADDRESS
S_PHONE        S_ACCTBAL
---------- ---------- ---------------------------------------------------------------------------------------- ------------------------- ---------------------------
-------- --------------- ----------
    300001         15 4Mg3kxM6RN0R xSNAM7zxC56w                                                                Supplier#000300001        Chyg6hnCBRNi
25-828-110-8355    8919.47
    300002          1 500 mBk02mMyQilwkigjyLRj7AMmNCLB2kh1L4AO1yNz5Q1P3NyA14PnAQxiSnL42jnBOzLPz04LyywC Supplier#000300002        Q0Q1MmP4RSxm z6
11-138-442-1426    -374.33
    300003          0 3P6kN0mxyzB 23SzhL5w0y xPAj7xk3Q7jA1ALhmR0N470l70i3l7AkLAjQS3wmzPmmgn06B4n0x O5M Supplier#000300003        R7BjCz0zx3QCgASl3 ANSwzmB
10-976-779-6099    9587.6
    300004          4 LM1nlCh 0w0hnnAmy7kOhA37wOj5OQ2mNSMmnQ5xiz41OwRAimP y75jz xSl1MQ5               Supplier#000300004        PA2 iCl3N4 0S2 h5 wLQwgg5yNOlhL3
14-405-780-1609    8818.61
    300005         11 CCOh5RS53P6yRjg41w2jwy6j6wgSCg0n2wkMnhx217zM                                           Supplier#000300005        j3AkMM P2 w0MMMkO il
21-455-171-8231    9418.59
    300006         12 6AkMz26BPAxMAlLhMyjBPC4M6 gS2ilglyONMNOxx55kwg                                         Supplier#000300006        Lkk2gSjx06j6Qwj4MkBO1AQL2RA
22-184-454-3494    6362.51
    300007         11 Lz7O0j5BjhClQAABziOCh3mC63A7xOn2CBOi                                                   Supplier#000300007        h3g3N76P4xQgkygmQ5
PkiBgn2S1zwyy43Ag1Bl  21-797-983-8665    1688.24
    300008          3 mBP7jmS j4C0LyRNRk3M6RQ6My0Mmw3ilMC5lw 1RjQiNmO6Bkm22MOS33knB                          Supplier#000300008        353gL6On5CzB2Aw2BA
13-744-817-2919    3175.37
    300009          4 62mQ63hRhLR2RMj15m0PSyi7SA4mLj2R2 MLxNj6mS6jShi6lgA16m1glA0Rg5y 31zy3k025RiOBkM5 Supplier#000300009        xRPM6hCmBOwMPlln0L531
14-453-567-5044    5392.67
    300010          0 3wkSCjPR1xBN3LP54Nwhwi1O SmBAzNglxPLnC6i65yBP qQwzOO7SjxQg wCM7y302kwAkwg        Supplier#000300010        LmjQSO6w3wyOz
10-620-450-5622    3609.34
10 rows selected.
PS_PARTKEY PS_SUPPKEY PS_SUPPLYC PS_AVAILQT PS_COMMENT
---------- ---------- ---------- ---------- ----------------------------------------------------------------------------------------
```

```
   6000001          2     515.12       1957 360A3ziywSLgyMmx2N10Qhzk1PM7lP1CmO 45i7nhO4lB NQQ1jMPm33Lwh0Rl0AnSM7QRy4xBAS35 R
   6000001     250008      537.4       2374 SkOn5n101O1C3M751Sygzwhmy307mlAPgxml3m0Mij7AnBh4BxO00A0Q0jP5n 2ORRNLkzg5gih27l3m
   6000001     500014      226.3       9601 l4iLh40yMjiMjgPQiw PhlwhSmhwRR67124MN4yyny 5LNQNQRnj1CRlmQC1Ah644B51MAnQO11ij3xz
   6000001     750020     432.58       7795 4ClCQRCzkyB2in4P6izRjOLk4iQ2QlL3QNkk314APBh6z6jh7BP7xwP46QNPLQ22xR24h5hPgmP
   6000002          3     817.48       4562 5C4QN14w2wmBmC1ywznQiLmO26in123C3yx5On3Sh55nm4mwCOBg7Ahjh0zSPSOj1wQhx7MCn3xC21ww
   6000002     250009     222.77        411 nLjwN6M5NzC6ym5 LmzMzAlO1hgmyQkj127l xM4g2PRmBLC7ykSn7wC51i6C6jxORy nii2CN4g0nmC
   6000002     500015      96.32       3245 LlNPQx7m36NMBl55R3032mx2N0g1Ow0AL5zzxygQj5AOS4Qy4z6Nj3nl27LiOLLS4kANSP2x PjRMkh7
   6000002     750021     766.04       5001 ln6n mlBQOAL ih6y24ig L5gAQ1zB60y5iA3khlA650 6NSO0Alj
   6000003          4     776.94       6415 1iS0Mk7CmiRPQ71ilPPSkmzLwxB w7mLPzC Pjw1iy6yhwSB33MP4QwC MNNPiMgxwg6nkzj xmySPiy
   6000003     250010     248.38       1192 7272SAj6m064mS0hNSjOABP Q0mxSB wzz4ihj257i0wL1l0LN jCmhwOLgnCljA0wC3Bhin4xh2xxmk
10 rows selected.
N_NATIONKE N_NAME                    N_REGIONKE N_COMMENT
---------- ------------------------- ---------- ---------------------------------------------------------------------------------------
         0 ALGERIA                             0 2Cxhl7 L1iwk6hMh300izngN32CPwCikyLk6khMzSRA
         1 ARGENTINA                           1 zQn3Okwz1wLn7PLS3OhCgn56kP5PyRikgi1B71L
         2 BRAZIL                              1 gLmS0nACAmnBCj2klki7RCPNgPxnCOjNg4k OiAg57COSOm1NwCnOyLx40R SC y20gPPAkNk5hxRhR5
         3 CANADA                              1  4yMO AhnQ5Lh wzQAM662Aw1ByCl7CxmzRwNR5nAlO4 x
         4 EGYPT                               4 l1im5126  Cxj NMQmLxOikni02j2m3Ah4yNR1QQiL507j2QSlyN
         5 ETHIOPIA                            0 NS7n LSOP Oz5n1AlB2S02nN0lMh4SBxP iRhBO 047R26 2BlM
         6 FRANCE                              3 3mjmizl S 3L3k2hNNhNlP4w370xRxyN15wn
         7 GERMANY                             3 z nOP4RkwO CmzBB 516mAg lByw4OM3QyNPA
         8 INDIA                               2 MNlR5RCiRMj1l11wjN7Myn M1lylNlMmBQl7PL4C  kkxQkgPQ7i3w6B67R2QkOO40xl4Q2iw76jRL7i
         9 INDONESIA                           2 SjPmQO71Lj 7ABj6MxlAQk3nLwi73BPxzCwjzMn4zlLzgg6nnz0j0w zxC66gP6ykRPMg
10 rows selected.
R_REGIONKE R_NAME                    R_COMMENT
---------- ------------------------- -------------------------------------------------------------------------------
         0 AFRICA                    xSx31zz31Cl1z4OAnmm05AjiOxC3AMMNOgC0kACgwngg3glP7LLLywlQy7R
         1 AMERICA                   kgyh3LSnC72k6zlAz0LP3k2L4QB1QL1O673OjO1SPj0ngQ7CO100SBgmgRQ4lgPCMk21A425iklyAR4y
         2 ASIA                      NSg6xlMlA1lzm6mOR0Ajx nhRA77NgRxBwL1M6Py RjySB3RLwkyPkwMM2R1BQ xAzkOgkjml10gAghi
         3 EUROPE                    zlSL7Qwg12hMBL5lhlz0M45QkjShwSyiO04MLOh7wn1ARLQPyPAyAii1576l1Li7AlnR1S RQ4SLny7B
         4 MIDDLE EAST               R1lxmhPLz3Cy2mNlg4QMBnNASM ACki MPki7Oi
5 rows selected.
```

Appendix G: Auditor’s Attestation Letter