Legend DeepComp 6800 Server

using

Oracle10g Database Enterprise Edittion

and

Red Hat- Enterprise Linux- AS

TPC Benchmark™ H
Full Disclosure Report

December 26, 2003

Second Edition - December 26, 2003

Printed in China, December 26, 2003.

Overview

This report documents the methodology and results of the TPC Benchmark. H test conducted on the Legend DeepComp 6800 Server, in conformance with the requirements of the TPC Benchmark. H Standard Specification, Revision 2.0. The operating system used for the benchmark was Redhat Linux Advanced Server for IA64; the DBMS was Oracle10g for IA64.

Standard and Executive Summary Statements

The pages following this preface contain the Executive Summary and Numerical Quantities Summary of the benchmark results.

Auditor

The benchmark configuration, environment and methodology used to produce and validate the test results and the pricing model used to calculate the cost per QphH was audited by Francois Raab, InfoSizing, to verify compliance with the relevant TPC specifications.

TPC Benchmark H Overview

The TPC Benchmark . H (TPC-H) is a decision support benchmark. It consists of a suite of business oriented ad-hoc queries and concurrent data modifications. The queries and the data populating the database have been chosen to have broad industry-wide relevance while maintaining a sufficient degree of ease of implementation. This benchmark illustrates decision support systems that:

- Examine large volumes of data;
- Execute queries with a high degree of complexity;
- Give answers to critical business questions.

TPC-H evaluates the performance of various decision support systems by the execution of sets of queries against a standard database under controlled conditions. The TPC-H queries:

- Give answers to real-world business questions;
- Simulate generated ad-hoc queries(e.g., via a point and click GUI interface);
- Are far more complex than most OLTP transactions;
- Include a rich breadth of operators and selectivity constraints;
- Generate intensive activity on the part of the database server component of the system under test;
- Are executed against a database complying to specific population and scaling requirements;
- Are implemented with constraints derived from staying closely synchronized with an on-line production database.

The TPC-H operations are modeled as follows:

- The database is continuously available 24 hours a day, 7 days a week, for ad-hoc queries from multiple end users and updates against all tables, except possibly during infrequent (e.g., once a month) maintenance sessions;
- The TPC-H database tracks, possibly with some delay, the state of the OLTP database through ongoing updates which batch together a number of modifications impacting some part of the

decision support database;

- Due to the world-wide nature of the business data stored in the TPC-H database, the queries and the updates may be executed against the database at any time, especially in relation to each other. In addition, this mix of queries and updates is subject to specific ACIDity requirements, since queries and updates may execute concurrently;
- To achieve the optimal compromise between performance and operational requirements the database administrator can set, once and for all, the locking levels and the concurrent scheduling rules for queries and updates.

The minimum database required to run the benchmark holds business data from 10,000 suppliers. It contains almost ten million rows representing a raw storage capacity of about 1 GB. Compliant benchmark implementations may also use one of the larger permissible database populations (e.g. 1000 GB), as defined in Clause 4.1.3.

The performance metrics reported by TPC-H measure multiple aspects of the capability of the system to process queries. The TPC-H metric at the selected size (QphH@Size) is the performance metric. To be compliant with the TPC-H standard, all references to TPC-H results for a given configuration must include all required reporting components (see Clause 5.4.7). The TPC believes that comparisons of TPC-H results measured against different database sizes are misleading and discourages such comparisons.

The TPC-H database must be implemented using a commercially available database management system (DBMS), and the queries executed via an interface using dynamic SQL. The specification provides for variants of SQL, as implementers are not required to have implemented a specific SQL standard in full. TPC-D uses terminology and metrics that are similar to other benchmarks, originated by the TPC and others. Such similarity in terminology does not in any way imply that TPC-H results are comparable to other benchmarks. The only benchmark results comparable to TPC-H are other TPC-H results compliant with the same revision.

Despite the fact that this benchmark offers a rich environment representative of many decision support systems, this benchmark does not reflect the entire range of decision support requirements. In addition, the extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-H approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to any other environment are not recommended.

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

Benchmark sponsors are permitted several possible system designs, provided that they adhere to the model described in Clause 6. A full disclosure report (FDR) of the implementation details, as specified in Clause 8, must be made available along with the reported results.

General Implementation Guidelines

The purpose of TPC benchmarks is to provide relevant, objective performance data to industry users. To achieve that purpose, TPC benchmark specifications require that benchmark tests be implemented with systems, products, technologies and pricing that:

- Are generally available to users;
- Are relevant to the market segment that the individual TPC benchmark models or represents (e.g. TPC-H models and represents complex, high data volume, decision support environments);
- Would plausibly be implemented by a significant number of users in the market segment the benchmark models or represents.

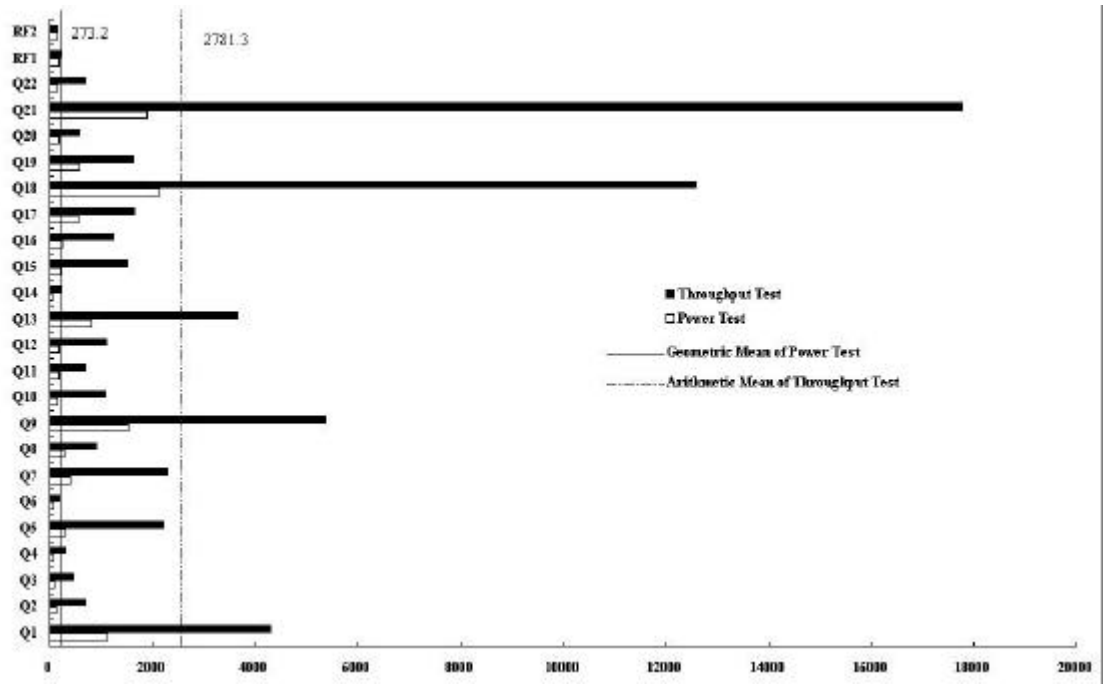
The performance metric reported by TPC-H is called the TPC-H Composite Query-per-Hour Performance Metric (QphH@Size), and reflects multiple aspects of the capability of the system to process queries. These aspects include the selected database size against which the queries are executed, the query processing power when queries are submitted by a single stream, and the query throughput when queries are submitted by multiple concurrent users. The TPC-H Price/Performance metric is expressed as RMB /QphH@Size.

The extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-H approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to any other environment are not recommended.

lenovo联想 ORACLE	**Legend DeepComp 6800 with Oracle Database 10g**	TPC-H Revision 2.0 Report Date: Nov. 6, 2003

Total System Cost	Composite Query per Hour Metric	Price/Performance
RMB 13,145,726	**9950.7 QphH@1000GB**	**RMB 1,321 per QphH@1000GB**

Database Size	Database Manager	Operating System	Other Software	Availability Date
1000GB	**Oracle Database 10g Enterprise Edition with Real Application Clusters and Partitioning**	**Red Hat- Enterprise Linux- AS (v.3 Standard for Itanium- Processor)**		**May 6, 2004**



Query times in seconds

Database Load Time = 6h44m	Load Includes Backup: N	Total Data Storage/Database Size = 39.43
RAID (Base Tables Only): N	RAID (Base Tables and Auxiliary Data Structures): N	RAID (All): Y

System Configuration

Processors:	16 x Itanium2 1.3GHz, 32KB L1-cache, 256KB L2-cache, 3MB L3-cache
Memory:	64 GB
Disk Drives:	4 x Legend SureFibre 920
Total Disk Storage:	39432.2 GB (In this calculation one GB is defined as 1024*1024*1024 bytes)
LAN Controllers:	8 x Intel Pro/1000 XT LAN Adapter
Ethernet Switch:	Legend Extreme 6808
Fibre Switch:	Legend Brocade BR3902

Note: Database Size includes only raw data (e.g., no temp, index, redundant storage space, etc).

lenovo联想 ORACLE	Legend DeepComp 6800 with Oracle Database 10g	TPC-H Revision 2.0 Report Date: Nov. 6, 2003

Description	Part Number	Source	Reference Price	Qty	Extended Price	3 year. Maintenance Price
Server Hardware						
Legend DeepComp 410 (Base System, not include CPU, MEM, HD and VRM)	888-004081	1	93,431	4	373,724	78,083
Intel Itanium2 1.3GHz CPU	11-005098	1	39,448	16	631,168	
73GB 15K RPM SCSI HDD	16-001091	1	9,000	4	36,000	
1GB DDR Memory	1002247	1	7,800	64	499,200	
VRM	11-004060	1	3,817	4	15,268	
Qlogic 2310F Host Bus Adapter	11-003520	1	16,180	16	258,880	
Intel Pro/1000 XT Lan Adapter	11-003532	1	2,200	8	17,600	
2M Ext.5 Copper Cable	---	1	15	12	180	
Legend USB Keyboard	25-001071	1	120	1	120	
Legend USB Mouse	25-000464	1	90	1	90	
Legend 15" Color Monitor	88-000174	1	1,200	1	1,200	
APC Symmetra Power Array 12KVA N+1	20-001231	1	118,647	1	118,647	
				Subtotal:	1,952,077	78,083
Storage						
Legend SureFibre 920R	---	1	1,130,872	4	4,523,488	752,322
Legend SureFibre 920J	16-001146	1	109,475	24	2,627,400	
146GB 10K RPM FC HDD	16-001123	1	32,600	288	9,388,800	
146GB 10K RPM FC HDD(10% spare)	16-001123	1	32,600	29	945,400	
2M LC-LC Fibre Optic Cable	310-09314	1	980	80	78,400	
Legend Brocade BR3902 Fiber Switch	---	1	1,141,772	1	1,141,772	
Legend Short Wave SFP	55-000641	1	3,212	32	102,784	
				Subtotal:	18,808,044	752,322
Connectivity						
Legend iSpirit 6808	53-000592	1	220,963	1	220,963	included
- 6808 G8Ti Model	53-000599	1	146,788	1	146,788	
- 6808 F48Ti Model	53-000604	1	174,328	1	174,328	
- 6808 MSM64i Model	53-000594	1	220,228	1	220,228	
				Subtotal:	762,307	

Hardware and maintance Discount						
Large volume discount on Legend hardware		1	51%	1	(10,976,438)	(423,506)
			Hardware Subtotal:		10,545,990	406,898
Server Software						
Red Hat Enterprise Linux AS for Itanium Processor (Version. 3 Stand Edition.)	31013839	3	15,000	4	60,000	
2 Additional. Years Subs. to Red Hat Linux AS for Itanium (Version. 3 Stand Edition.)	31013839	3	15,000	8		120,000
Oracle Database 10g Enterprise Edition for 3 years, Named User Plus		2		16	1,324,200	
Real Application Clusters for 3 years, Named User Plus		2		16	662,200	
Partitioning for 3 years, Named User Plus		2		16	331,000	
Oracle Database Server Support Package for 3 years		2		3	198,648	
				Subtotal:	2,576,048	120,000
Oracle Mandatory E-Business Discount		2			(503,210)	
			Software Subtotal:		2,072,838	120,000
				Total:	12,618,828	526,898
				Three-Year Cost of Ownership: RMB		13,145,726

Pricing: 1 - Legend; 2 - Oracle (Pricing Contact: MaryBeth Pierantoni, mary.beth.pierantoni@oracle.com, 650-506-2118); 3 - Redhat, OEMed by Legend	**QphH@1000GB:**	9950.7
Warranty and Maintenance: The standard warranty has been upgraded to 3 years of 24x7x4 coverage.	**RMB/QphH@1000GB: RMB**	1,321
*All discounts are based on China Yuan(RMB) list prices and for similar quantities and configurations.		

Audited By: Francois Raab for InfoSizing (www.sizing.com)

Prices used in TPC benchmarks reflect actual prices a customer would pay for a one-time purchase of the stated components. Individually negotiated discounts are not permitted. Special prices based on assumptions about past or future purchases are not permitted. All discounts refelect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark specifications.

If you find the stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank you.

Measurement Results:

Database Scaling (SF/size)	1000
Total Data Storage/Database Size	39.43
Start of Database Load Time	2003-10-28 05:01:06
End of Database Load Time	2003-10-28 11:46:19
Database Load Time	6h46m
Query Streams for Throughput Test (S)	7
TPC-H Power	13178.3
TPC-H Throughput	7513.6
TPC-H Composite Query-per-Hour Metric (QphH@1000GB)	9950.7
Total System Price Over 3 Years (RMB)	13,145,726
TPC-H Price/Performance Metric (RMB/QphH@1000GB)	1,321

Measurement Intervals:

Measurement Interval in Throughput Test (Ts)	73786

Duration of Stream Execution:

	SEED	Start Date/Time	End Date/Time	Duration
Stream 00	1028120446	2003-10-28 12:21:35	2003-10-28 15:37:17	3:15:42
Stream 01	1028120446	2003-10-28 15:37:21	2003-10-29 11:21:51	19:44:20
Stream 02	1028120446	2003-10-28 15:37:21	2003-10-29 09:50:59	18:13:28
Stream 03	1028120446	2003-10-28 15:37:21	2003-10-29 09:06:00	17:28:29
Stream 04	1028120446	2003-10-28 15:37:21	2003-10-29 07:20:29	15:42:58
Stream 05	1028120446	2003-10-28 15:37:21	2003-10-29 10:56:06	19:18:45
Stream 06	1028120446	2003-10-28 15:37:21	2003-10-29 08:13:04	16:35:33
Stream 07	1028120446	2003-10-28 15:37:21	2003-10-29 03:31:33	11:54:02
Refresh	1028120446	2003-10-28 15:37:21	2003-10-29 12:07:07	20:29:36

TPC-H Timing Intervals (in seconds)

Stream ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Stream 00	1131.2	127.5	92.5	67.8	305.4	64.6	411.0	282.5
Stream 01	1124.2	476.2	378.7	72.2	1611.7	90.0	1231.7	515.7
Stream 02	5709.1	1154.3	417.0	344.0	1666.2	122.9	1793.8	1146.3
Stream 03	6402.1	673.7	457.9	581.0	3753.6	234.0	3267.1	706.7
Stream 04	3433.5	712.5	550.0	303.5	1013.3	132.4	2402.4	1132.3
Stream 05	3133.1	989.4	337.5	310.1	1628.6	81.5	2194.7	653.3
Stream 06	6199.9	559.7	491.5	340.8	2677.9	321.7	2718.9	1599.5
Stream 07	4169.4	287.2	578.4	236.1	3180.4	292.1	2556.4	617.4
Minimum	1124.2	287.2	337.5	72.2	1013.3	81.5	1231.7	515.7
Average	4310.2	693.3	458.7	312.5	2218.8	182.1	2309.3	910.2
Maximum	6402.1	1154.3	578.4	581.0	3753.6	321.7	3267.1	1599.5

Stream ID	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16
Stream 00	1537.1	164.5	188.4	188.2	818.3	36.7	200.9	239.1
Stream 01	1589.2	432.5	257.5	334.1	1376.8	53.8	367.8	406.4
Stream 02	7934.4	2497.3	819.0	1107.4	3576.2	169.4	1337.0	1518.0
Stream 03	6154.9	1185.0	643.4	1828.3	3616.7	88.4	1693.9	918.3
Stream 04	5227.1	966.4	589.3	1304.6	5137.5	768.2	1227.4	1112.6
Stream 05	2231.1	353.0	1018.8	369.9	3351.9	112.1	1413.4	1045.6
Stream 06	8329.6	1248.2	720.9	1376.3	4590.6	76.9	2222.2	1320.5
Stream 07	6100.0	949.9	819.2	1416.8	3907.4	234.4	2259.8	2353.5
Minimum	1589.2	353.0	257.5	334.1	1376.8	53.8	367.8	406.4
Average	5366.6	1090.3	695.4	1105.3	3651.0	214.7	1503.1	1239.3
Maximum	8329.6	2497.3	1018.8	1828.3	5137.5	768.2	2259.8	2353.5

Stream ID	Q17	Q18	Q19	Q20	Q21	Q22	RF1	RF2
Stream 00	562.6	2149.2	592.1	181.3	1904.6	168.8	173.8	153.1
Stream 01	791.0	13451.1	887.3	296.9	45157.7	167.8	199.7	165.7
Stream 02	5370.5	13149.9	3363.8	829.8	10962.1	629.3	209.8	166.2
Stream 03	1737.7	14327.0	2384.3	822.1	10918.7	524.3	190.4	166.3
Stream 04	731.4	13637.0	1161.3	900.1	13081.0	1064.7	236.7	162.9
Stream 05	644.8	14583.7	1611.6	243.3	32401.4	815.9	245.2	165.3
Stream 06	726.8	12640.7	1045.6	647.8	8760.7	1126.2	233.6	166.4
Stream 07	1540.7	6355.6	963.0	337.4	3088.0	608.1	231.5	176.4
Minimum	644.8	6355.6	887.3	243.3	3088.0	167.8	190.4	162.9
Average	1649.0	12592.1	1631.0	582.5	17767.1	705.2	221.0	167.0
Maximum	5370.5	14583.7	3363.8	900.1	45157.7	1126.2	245.2	176.4

INFO SIZING

TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL CERTIFIED AUDITOR

Test Sponsors:

Ray Glasstone
Manger, DSS Performance
Oracle Corporation
100 Oracle Parkway
Redwood Shores, CA 94065

Feng Rui
SuperServer & Services Business Dept.
Legend Group Limited
No. 32 Chuang Ye Middle Road
Haidian District, Beijing, China

November 2, 2003

I verified the TPC Benchmark™ H performance of the following configuration:

Platform: **Legend DeepComp 6800**

Database Manager: **Oracle Database 10g Enterprise Edition**

Operating System: **RedHat Linux AS 3.0 for IA64**

The results were:

CPU (Speed)	Memory	Disks	**QphH@1000GB**
Legend DeepComp 6800			
16 x Intanium2 (1.3 GHz)	3 MB L3-Cache/cpu 64 GB Main	288 x 146 GB 4 x 73 GB	**9,950.7**

In my opinion, this performance result was produced in compliance with the TPC's requirements for the benchmark. The following verification items were given special attention:

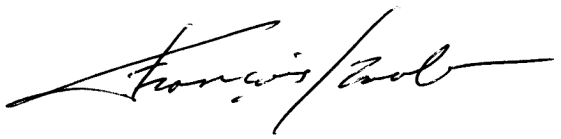
- The database records were defined with the proper layout and size
- The database population was generated using DBGEN
- The database was properly scaled to 1 TB and populated accordingly
- The compliance of the database auxiliary data structures was verified
- The database load time was correctly measured and reported

- The required ACID properties were verified and met
- The query input variables were generated by QGEN
- The query text was produced using minor modifications and no variant
- The execution of the queries against the SF1 database produced compliant answers
- A compliant implementation specific layer was used to drive the tests
- The throughput tests involved 7 query streams
- The ratio between the longest and the shortest query was such that no query timing was adjusted
- The execution times for queries and refresh functions were correctly measured and reported
- The repeatability of the measured results was verified. A failure during the second run of the benchmark required the execution of a third run, from which the reported results were collected.
- At least 8 hours of database log was configured
- The system pricing was verified for major components and maintenance
- The major pages from the FDR were verified for accuracy

Additional Audit Notes:

None.

Respectfully Yours,



François Raab
President

1 General Items

1.1 Benchmark Sponsor

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

This TPC-H benchmark is sponsored by Legend Group Limited and Oracle Corporation.

1.2 Parameter Settings

Settings must be provided for all customer-tunable parameters and options that 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 parameters*
- *Configuration parameters and options for any other software component incorporated into the pricingstructure*
- *Compiler optimization options*

Appendix A contains the operating system and Oracle parameters used in this benchmark.

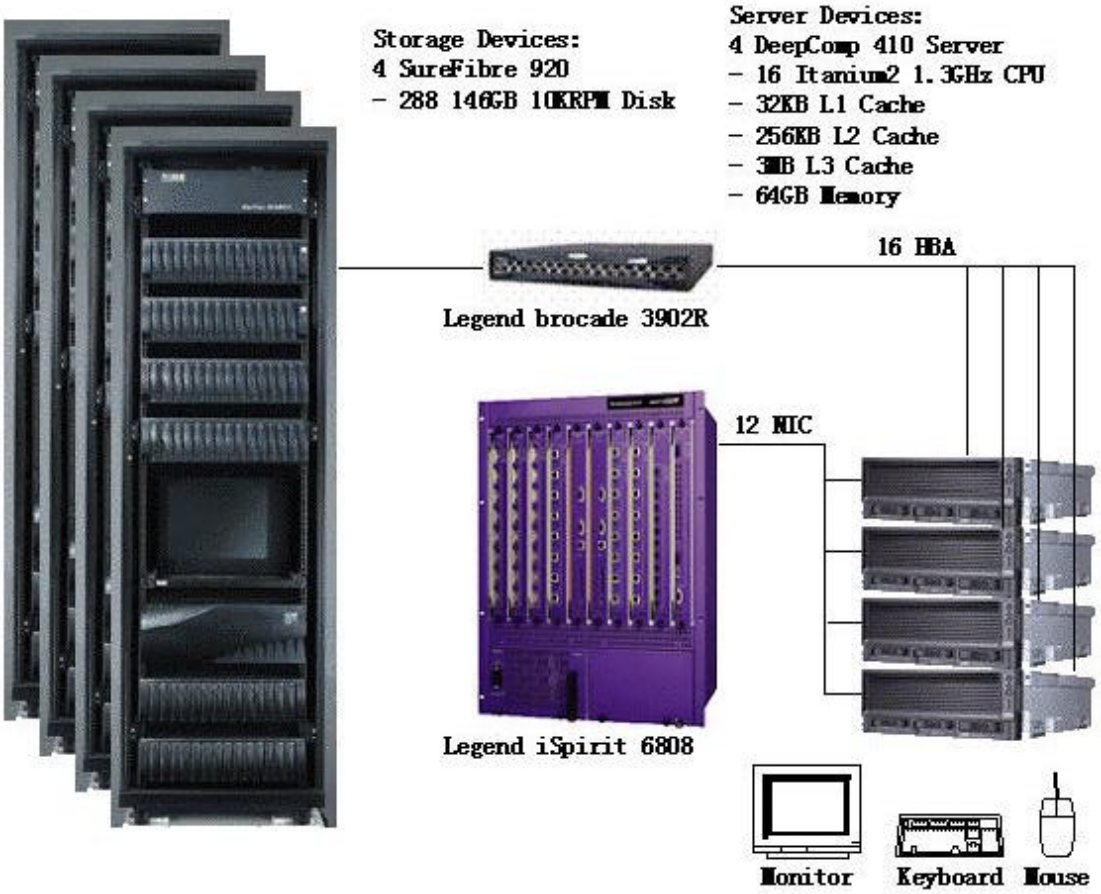
1.3 Configuration Diagram

Provide diagrams of both the measured and priced configurations, accompanied by a description of the differences.

DeepComp 6800, measured and priced configuration:

- 16 Itanium2 1.3 GHz processors (3MB L3 cache)
- 64 GB memory
- 4 * 73 GB internal disks
- 288 * 146 GB disks
- 4 * SureFibre 920 Storage Array
- 16 * Qlogic 2310F Fibre Channel controllers
- 1 * Legend iSpirit 6800 Ethernet Switch
- 8 * Intel Pro/1000 XT Adapters
- 1 * Legend Brocade 3902R Fibre Switch

There is no difference between the measured configuration and priced configuration. The diagram is the same for both the priced and measured configurations.


Storage Devices:
4 SureFibre 920
- 288 146GB 10KRPM Disk
Server Devices:
4 DeepComp 410 Server
- 16 Itanium2 1.3GHz CPU
- 32KB L1 Cache
- 256KB L2 Cache
- 3MB L3 Cache
- 64GB Memory
16 HBA
Legend brocade 3902R
12 NIC
Legend iSpirit 6808
Monitor
Keyboard
Mouse

2 Clause 1 Logical Database Design

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 programs and scripts that create and analyze the tables and indexes for the TPC-H database.

2.1 Database Definition Statements

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

Appendix B describes the scripts that define, create, and analyze the tables and indices for the TPC-H database.

2.2 Physical Organization

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. Column ordering was reordered in tables. Refer to the table create statements in Appendix B for further details.

2.3 Horizontal Partitioning

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

Horizontal partitioning was used for all base and index tables except NATION and REGION. The details of this partitioning can be understood by examining the syntax of the table and index definition statements in Appendix B. Similar partitioning was used in the qualification database size.

Section 5.2 describes the distribution of tables and logs across all media.

2.4 Replication

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

No replication was used.

3 Clause 2 Queries and Refresh Functions

3.1 Query Language

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

SQL was the query language used to implement all queries.

3.2 Verifying Method for Random Number Generation

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

TPC supplied versions 1.3.0 of DBGEN and QGEN were used for this TPC-H benchmark.

3.3 Generating Values for Substitution Parameters

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.

QGEN version 1.3.0 was used to generate the substitution parameters.

3.4 Query Text and Output Data from Qualification 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 (seeClause 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 D contains the qualification query text and query output.

3.5 Query Substitution Parameters and Seeds Used

The query substitution parameters used for all performance tests must be disclosed in tabular format, along with the seeds used to generate these parameters.

Appendix E contains the seed and query substitution parameters.

3.6 Query Isolation Level

The isolation level used to run the queries must be disclosed. If the isolation level does not map closely to the levels defined in Clause 3.4, additional descriptive detail must be provided.

The queries and transactions were run with isolation level 3 (repeatable read).

3.7 Source Code of Refresh Functions

The details of how the refresh functions were implemented must be disclosed (including source code of any noncommercial program used).

The refresh function is part of the driver code included in Appendix F.

4 Clause 3 Database System Properties

4.1 ACID Properties

The ACID (Atomicity, Consistency, Isolation and Durability) properties of transaction processing systems must be supported by the system under test during the timed portion of this benchmark. Since TPC-H is not a transaction processing benchmark, the ACID properties must be evaluated outside the timed portion of the test.

Source code for the ACID test is included in Appendix C.

4.2 Atomicity

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

4.2.1 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 ORDERS, LINEITEM, and HISTORY tables

1. The total price from the ORDERS 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 committed.
4. The total price from the ORDERS 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.

4.2.2 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 ORDERS, LINEITEM, and HISTORY tables.

1. The total price from the ORDERS 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 ORDERS table and the extended price from the LINEITEM table were retrieved for the same order key. It was verified that the appropriate rows had not been changed.

4.3 Consistency

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

4.3.1 Consistency Test

Verify that ORDERS 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 ORDERS and LINEITEM.

1. The consistency of the ORDERS and LINEITEM tables was verified based on a sample of order keys.
2. 100 ACID Transactions were submitted by each of nine execution streams.
3. The consistency of the ORDERS and LINEITEM tables was re-verified.

4.4 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 the proper order.

4.4.1 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 was suspended prior to COMMIT.
2. An ACID Query was started for the same O_KEY used in step 1. The ACID Query blocked and did not see the uncommitted changes made by the ACID Transaction.
3. The ACID Transaction was resumed and COMMITTED.
4. The ACID Query completed. It returned the data as committed by the ACID Transaction.

4.4.2 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 did not see the uncommitted changes made by the ACID Transaction.
3. The ACID Transaction was ROLLED BACK.
4. The ACID Query completed.

4.4.3 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. T1 was suspended prior to COMMIT.
2. Another ACID Transaction, T2, was started using the same O_KEY and L_KEY and a randomly

selected DELTA.
3. T2 waited.
4. T1 was allowed to COMMIT and T2 completed.
5. It was verified that T2.L_EXTENDEDPRICE = T1.L_EXTENDEDPRICE + (DELTA1*(T1.L_EXTENDEDPRICE/T1.L_QUANTITY))

4.4.4 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. T1 was suspended prior to ROLLBACK.
2. Another ACID Transaction, T2, was started using the same O_KEY and L_KEY and a randomly selected DELTA.
3. T2 waited.
4. T1 was allowed to ROLLBACK and T2 completed.
5. It was verified that T2.L_EXTENDEDPRICE = T1.L_EXTENDEDPRICE.

4.4.5 Concurrent Progress of Read and Write Transactions

Demonstrate the ability of read and write transactions affecting different database tables to make progress concurrently.

1. An ACID Transaction, T1, was started for a randomly selected O_KEY, L_KEY, and DELTA. T1 was suspended prior to ROLLBACK.
2. Another Transaction, T2, was started which did the following:
For random values of PS_PARTKEY and PS_SUPPKEY, all columns of the PARTSUPP table for which PS_PARTKEY and PS_SUPPKEY are equal, are returned.
3. T2 completed.
4. T1 was allowed to COMMIT.
5. It was verified that appropriate rows in ORDERS, LINEITEM and HISTORY tables were changed.

4.4.6 Read-Only Query Conflict with Update Transaction

Demonstrate that the continuous submission of arbitrary (read-only) queries against one or more tables of the database does not indefinitely delay update transactions affecting those tables from making progress.

1. A Transaction, T1, executing Q1 against the qualification database, was started using a randomly selected DELTA.
2. An ACID Transaction T2, was started for a randomly selected O_KEY, L_KEY and DELTA.
3. T2 completed and appropriate rows in the ORDERS, LINEITEM and HISTORY tables had been changed.
4. Transaction T1 completed executing Q1.

4.5 Durability

The SUT 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 3.5.3.

4.5.1 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-H database tables or recovery log tables.
The LUNs containing TPC-H data files and redo log files were pairs of physical disks mirrored by hardware (RAID1). Each TPC-H data file and redo log file was striped across all available LUNs by software (RAID0), i.e. each LUN contained pieces from all TPC-H data files and redo log files. During the durability test, a LUN (i.e. a mirrored pair of physical disks) was randomly chosen and one of its disks removed from the cabinet. The test continued uninterupted, using the remaining side of the mirror.

4.5.2 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.
The system crash and memory failure tests were combined. Powered down one of the 4 nodes. When power was restored, the system rebooted and the database was restarted. The durability success file and the HISTORY table were compared successfully.

4.5.3 Memory Failure

Guarantee the database and committed updates are preserved across failure of all or part of memory (loss of contents).
See section 4.5.2.

4.5.4 Ethernet switch Failure

Guarantee the database and committed updates are preserved across failure of all or part of memory (loss of connection).
Powered down the Etherent switch. When power was restored, the database was restarted. The durability success file and the HISTORY table were compared successfully.

4.5.5 Fibre switch Failure

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

Powered down the Fibre switch. When power was restored, reloaded the driver of Host Bus Adapters and re-recognized the storage devices, and then the database was restarted. The durability success file

and the HISTORY table were compared successfully.

5 Clause 4 Scaling and Database

5.1 Ending Cardinality of Tables

The cardinality (i.e., 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.

Table	Rows
Orders	1,500,000,000
Lineitem	5,999,989,709
Customer	150,000,000
Part	200,000,000
Supplier	10,000,000
Partsupp	800,000,000
Nation	25
Region	5

5.2 Distribution of Tables and Logs Across Media

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

There were 4 Legend SureFibre 920 storage arrays with 72 physical disks (installed in 6 SureFibre 920J Disk Enclosure) each. For each of these SureFibre 920, 9 Volume Groups (there was only one LUN on each volume group, which was recognized as a separate HDD by the OS) were created from pairs of mirrored physical disks and shown to Redhat Linux AS. All of the 4 SureFibre 920 were connected to a Legend Brocade BR3902 Fibre swith (32 ports). There were 4 Legend DeepComp 410 Servers in the system, each of which was connected to the Fibre switch by 4 Qlogic 2310F Host Bus Adapters.. The write caches of the SureFibre 920 were disabled.

There were 36 LUNs were available in total, and they were used as 2 different groups. The first group consisted of 4 LUNs, taking 1 LUN from each of the 4 SureFibre 920. This disk group was used for the dbgen flat files exclusively, not for the TPC-H database.

The second disk group consisted of the remaining 32 LUNs. There are 8 partitions created on each of these 32 LUNs, each of which was bound as a raw device to hold the Oracle data files and log files. For each of the Oracle data files and redo log files as listed in the database schema specification of Appendix B.

5.3 Database partition/replication mapping

The mapping of database partitions/replications must be explicitly described.

The database was not replicated.

Horizontal partitioning was used for base tables LINEITEM, ORDERS, PARTSUPP, PART, SUPPLIER and CUSTOMER. The details for this partitioning can be understood by examining the syntax of the table and index definition statements in Appendix B.

5.4 RAID Feature

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 must be disclosed for each device.

Table/Index	RAID type
tables	RAID 1+0
indexes	RAID 1+0
temp tablespace	RAID 1+0
log	RAID 1+0
system tablespace	RAID 1+0

5.5 Modifications to the DBGEN

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 1.3.0 was used to generate the database population for this benchmark.

5.6 Database Load Time

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

The database load time was 6 hours 46 minutes.

5.7 Data Storage Ratio

The data storage ratio must be disclosed. It is computed as the ratio between the total amount of priced disk space, and the chosen test database size as defined in Clause 4.1.3.

The data storage ratio is computed from the following information:

Disk Type	# Of Disks	Space Per Disk*	Sub-Total Disk Space**
internal	4	73.0 GB	271.95 GB
SureFibre 920	288	146.0 GB	39,160.25 GB
		Total Space	**39,432.20 GB**
		Scale Factor	**1000**
		Data Storage Ratio	**39.43**

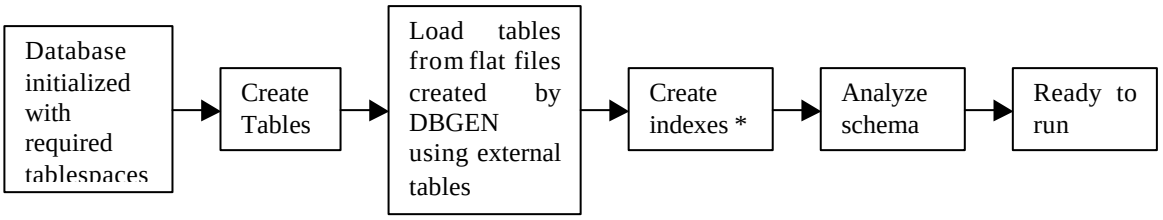
***Disk manufacturer definition of one GB is 10^9 byte*

****In this calculation one GB is defined as 2^30 bytes*

5.8 Database Load Mechanism Details and Illustration

The details of the database load must be described, including a block diagram illustrating the overall process.

The database was loaded using data generation stored on flat files all on the tested and priced configurations. Oracle created external tables using the files that were created by the DBGEN program.



*Analyze index performed during index creation

5.9 Qualification Database Configuration

Any differences between the configuration of the qualification database and the test database must be disclosed.

The qualification database used identical scripts to create and load the data with changes to adjust for the database scale factor.

6 Clause 5 Performance Metrics and Execution Rules

6.1 System Activity Between Load and Performance Tests

Any system activity on the SUT that takes place between the conclusion of the load test and the beginning of the performance test must be fully disclosed.

Auditor requested queries were run against the database to verify the correctness of the load. All scripts and queries used are included in Appendix F.

6.2 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. RF1 Refresh Transaction
2. Stream 00 Execution
3. RF2 Refresh Transaction

6.3 Timing Intervals for Each Query and Refresh Functions

The timing intervals for each query and for both refresh functions must be reported for the power test.

The timing intervals for each query and both update functions are given in the Numerical Quantities Summary earlier in this document.

6.4 Number of Streams for the Throughput Test

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

7 streams were used for the throughput test.

6.5 Start and End Date/Times for Each Query Stream

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

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

6.6 Total Elapsed Time of the Measurement Interval

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

The total elapsed time of the throughput test is given in the Numerical Quantity Summary earlier in this document.

6.7 Refresh Function Start Date/Time and Finish Date/Time

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

The start and finish times for each refresh function in the refresh stream are given in the Numerical Quantity Summary earlier in this document.

6.8 Timing Intervals for Each Query and Each Refresh Function for Each Stream

The timing intervals for each query of each stream and each refresh function must be reported for the throughput test.

The timing intervals for each query and each refresh function for the throughput test are given in the Numerical Quantity Summary earlier in this document.

6.9 Performance Metrics

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

The performance metrics, and the numbers on which they are based, are given in the Numerical Quantity Summary earlier in this document.

6.10 The Performance Metric and Numerical Quantities from Both Runs

The performance metric and numerical quantities from both runs must be disclosed.

Performance results from the first two executions of the TPC-H benchmark indicated the following percent difference for the metric points:

Run ID	QppH@1000GB	QthH@1000GB	QphH@1000GB
Run 1	13178.3	7513.6	9950.7
Run 2	13102.8	7594.1	9975.2
Difference	- 0.6%	+ 1.1%	+ 0.2%

6.11 System Activity Between Performance Tests

Any activity on the SUT that takes place between the conclusion of Run1 and the beginning of Run2 must bedisclosed.

There was no activity on the SUT between run1 and run2.

7 Clause 6 SUT and Driver Implementation

7.1 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.

All stream executions are performed by a single script. QGEN is used to produce query text.

For each power-test run:

- The SQL for RF1 is submitted to the database
- Then the queries as generated by QGEN are submitted in the order defined by Clause 5.3.5.4
- The SQL for RF2 is submitted to the database.

7.2 Implementation-Specific Layer

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.

The source code for the "qexec" Utility can be found in Appendix F.

7.3 Profile-Directed Optimization

If profile-directed optimization as described in Clause 5.2.9 is used, such use must be disclosed. Profiledirected optimization was not used.

Profile-directed optimization subject to the requirements of *5.2.9* and *5.2.10* was not used

8 Clause 7 Pricing

8.1 Hardware and Software Used

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 hardware and software used in the priced system is included in the pricing sheet in the Executive Summary. The price quotations are included in Appendix G.

8.2 Total Three Year Price

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

A detailed pricing sheet of all hardware and software used in this configuration and the 3-year maintenance costs, demonstrating the computation of the total 3-year price of the configuration, is included in the Executive Summary. The price quotations are included in Appendix G.

8.3 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 reported availability date for the priced system must be the date at which all components are committed to be available.

All hardware and software components will be available May 6, 2004.

8.4 Country-Specific Pricing

Additional Clause 7 related items may be included in the Full Disclosure Report for each country-specific priced configuration. Country-specific pricing is subject to Clause 7.1.7.

The configuration is priced for the China.

9 Auditor's Information and Attestation Letter

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. This implementation of the TPC Benchmark H was audited by Francois Raab for InfoSizing. Further information regarding the audit process may be obtained from:

Francois Raab
InfoSizing
francois@sizing.com
1373 N. Franklin St. Colorado Springs, CO 80903
(719) 473-7555
(719) 473-7554

The auditor’s attestation letter is included at the front of this report.

Appendix A Parameter Settings

A.1 init_run.ora

```
aq_tm_processes =0
statistics_level = BASIC
audit_trail = FALSE
compatible = 10.0.0.0
control_files = (/home/oracle/dev/raw/cntrl_1, /home/oracle/dev/raw/cntrl_2)
db_block_checksum = false
db_block_size = 8192
db_cache_size = 1g
db_file_multiblock_read_count = 128
db_files = 512
db_name = 10i
db_writer_processes = 4
dml_locks = 40000
enqueue_resources = 40000
global_names = FALSE
instance_name = rac1
shared_pool_size = 512000000
large_pool_size = 500m
log_buffer = 4194304
log_checkpoints_to_alert = true
max_dump_file_size = unlimited
nls_date_format = YYYY-MM-DD
open_cursors = 600
optimizer_index_cost_adj = 25
optimizer_mode = CHOOSE
optimizer_features_enable = 10.0.0.1
parallel_adaptive_multi_user = TRUE
parallel_execution_message_size = 8192
parallel_max_servers = 128
parallel_min_servers = 128
pga_aggregate_target = 8g
processes = 500
query_rewrite_enabled = true
recovery_parallelism = 8
replication_dependency_tracking = false
sessions = 384
transactions = 10
undo_management = auto
undo_retention =300000
```

A.2 init_rac1.ora

```
instance_number =1
thread =1
undo_management = auto
undo_retention = 3
UNDO_TABLESPACE = ts_undo1
cluster_database = true
#cluster_interconnects = 192.168.2.1:192.168.3.1
cluster_interconnects = 192.168.3.1
ifile=$ORACLE_HOME/dbs/init_run.ora
```

A.3 init_rac2.ora

```
instance_number =2
thread =2
undo_management = auto
undo_retention = 3
UNDO_TA BLESPACE = ts_undo2
cluster_database = true
#cluster_interconnects = 192.168.2.2:192.168.3.2
cluster_interconnects = 192.168.3.2
ifile=$ORACLE_HOME/dbs/init_run.ora
```

A.4 init_rac3.ora

```
instance_number =3
thread    =3
undo_management                 = auto
undo_retention                  = 3
UNDO_TABLESPACE                 =
ts_undo3
cluster_database                = true
#cluster_interconnects          =
192.168.2.3:192.168.3.3
cluster_interconnects           = 192.168.3.3
ifile=$ORACLE_HOME/dbs/init_run.ora
```

A.5 init_rac4.ora

```
instance_number =4
thread    =4
undo_management                 = auto
undo_retention                  = 3
UNDO_TABLESPACE                 =
ts_undo4
cluster_database                = true
#cluster_interconnects          =
192.168.2.4:192.168.3.4
cluster_interconnects           = 192.168.3.4
ifile=$ORACLE_HOME/dbs/init_run.ora
```

A.6 .bashrc

```
# .bashrc

# User specific aliases and functions

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi
ulimit -c unlimited
export ORACLE_HOME=/home/oracle
export O=$ORACLE_HOME
export ORA_CRS_HOME=$ORACLE_HOME
export ORACLE_SID=`uname -n`
#export ORACLE_SID=`uname -n`_1gb

export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:
$ORACLE_HOME/lib:$PRACLE_HOME/rdb
ms/lib:$ORACLE_HOME/opsm/lib:$ORACLE
_HOME/opsm/bin:/lib:/usr/lib:/usr/local/lib
export FRAME_PATH=/oracle/kit/frame
export KIT_DIR=/oracle/kit/kit
export
PATH=.:$PATH:$ORACLE_HOME/bin:$FRA
ME_PATH/bin

#. $KIT_DIR/env

alias cdlog="cd $ORACLE_HOME/rdbms/log"
alias cdbs="cd $ORACLE_HOME/dbs"
alias cdkit="cd $KIT_DIR"
alias cdwork="cd $KIT_DIR/neil_work"
alias cdfr="cd $FRAME_PATH"
alias cdtest="cd $KIT_DIR/audit/tests"
alias cdmq="cd $KIT_DIR/myqueries"
alias cdstats="cd $FRAME_PATH/stats"
alias ltt="ls -ltr|tail"
alias cdacid="cd $KIT_DIR/acid"
```

Appendix B. Build Programs and Scripts

B.1 1TB_final.dat

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

%b-preproc

*sql
\echo "{}" > script*getenv(BUMPX_CTR).sql
\sqlplus /NOLOG <<!
\set echo on;
\set timing on;
\set termout on;
\connect / as sysdba;
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now from dual;
\@script*getenv(BUMPX_CTR).sql;
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now from dual;
\exit;
\!
\/bin/rm script*getenv(BUMPX_CTR).sql;


*load1
\sqlldr {}

*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-dbcre
*bgon=1
########################################
########################################
# Database Creation Phase
*sql
{
shutdown abort;
}
*wait
# creating database  and undo tablespace
*sql
{
startup pfile= /home/oracle/dbs/init_run.ora nomount;
create database
  controlfile reuse
   logfile '/home/oracle/dev/raw/log_1' size 4096m reuse,
           '/home/oracle/dev/raw/log_2' size 4096m reuse
   datafile '/home/oracle/dev/raw/sys_1' size 1024m reuse
   sysaux datafile '/home/oracle/dev/raw/sysaux_1' size 1024m reuse
  undo tablespace ts_undo1
        datafile '/home/oracle/dev/raw/undo_1' size 30720m reuse
  default temporary tablespace ts_temp
        tempfile '/home/oracle/dev/raw/temp_1' size 22272m reuse
        extent management local uniform size 10m
  maxdatafiles 4000
  maxinstances 4
;
}
*wait
```

```
*sql
{
create  undo tablespace ts_undo2
  datafile  '/home/oracle/dev/raw/undo_2'  size
30720m reuse
;
}
*sql
{
create  undo tablespace ts_undo3
  datafile  '/home/oracle/dev/raw/undo_3'  size
30720m reuse
;
}
*sql
{
create  undo tablespace ts_undo4
  datafile  '/home/oracle/dev/raw/undo_4'  size
30720m reuse
;
}
# creating extra logfile threads for rac 8 nodes
*sql
{
alter database add logfile thread 2
  '/home/oracle/dev/raw/log_3'  size  4096m
reuse,
  '/home/oracle/dev/raw/log_4'  size  4096m
reuse;
alter database enable public thread 2;
}
*sql
{
alter database add logfile thread 3
  '/home/oracle/dev/raw/log_5'  size  4096m
reuse,
  '/home/oracle/dev/raw/log_6'  size  4096m
reuse;
alter database enable public thread 3;
}
*sql
{
alter database add logfile thread 4
  '/home/oracle/dev/raw/log_7'  size  4096m
reuse,
  '/home/oracle/dev/raw/log_8'  size  4096m
reuse;
alter database enable public thread 4;
}
# building data dictionary
*sql
{
set termout off
set echo off
spool /tmp/cat
@?/rdbms/admin/catalog.sql;
@?/rdbms/admin/catparr.sql;
@?/rdbms/admin/catproc.sql;
connect system/manager
@?/rdbms/admin/utlxplan.sql;
@?/sqlplus/admin/pupbld.sql;
spool off
}
*wait
*bgoff
%e-dbcre
%b-sctso
*bgon=300
########################################
########################################
# Schema Creation Phase - datafiles only (no
tables or users)
# creating data tablespaces, datafiles
# creating tpch's ts_one tablespace

*sql
{
--drop tablespace ts_default including contents;
create tablespace ts_default
datafile  '/home/oracle/dev/raw/default_1'  size
1024m reuse
extent management local
autoallocate
;
}
*sql
{
--drop tablespace ts_l1 including contents;
```

```
create tablespace ts_l1
datafile '/home/oracle/dev/raw/line_1' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l2 including contents;
create tablespace ts_l2
datafile '/home/oracle/dev/raw/line_2' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l3 including contents;
create tablespace ts_l3
datafile '/home/oracle/dev/raw/line_3' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l4 including contents;
create tablespace ts_l4
datafile '/home/oracle/dev/raw/line_4' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l5 including contents;
create tablespace ts_l5
datafile '/home/oracle/dev/raw/line_5' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l6 including contents;
create tablespace ts_l6
datafile '/home/oracle/dev/raw/line_6' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l7 including contents;
create tablespace ts_l7
datafile '/home/oracle/dev/raw/line_7' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l8 including contents;
create tablespace ts_l8
datafile '/home/oracle/dev/raw/line_8' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*wait
*sql
{
```

```
--drop tablespace ts_l9 including contents;
create tablespace ts_l9
datafile '/home/oracle/dev/raw/line_9' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l10 including contents;
create tablespace ts_l10
datafile '/home/oracle/dev/raw/line_10' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l11 including contents;
create tablespace ts_l11
datafile '/home/oracle/dev/raw/line_11' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l12 including contents;
create tablespace ts_l12
datafile '/home/oracle/dev/raw/line_12' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l13 including contents;
create tablespace ts_l13
datafile '/home/oracle/dev/raw/line_13' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l14 including contents;
create tablespace ts_l14
datafile '/home/oracle/dev/raw/line_14' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l15 including contents;
create tablespace ts_l15
datafile '/home/oracle/dev/raw/line_15' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxe xtents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l16 including contents;
create tablespace ts_l16
datafile '/home/oracle/dev/raw/line_16' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*wait
*sql
```

```
{
--drop tablespace ts_l17 including contents;
create tablespace ts_l17
datafile '/home/oracle/dev/raw/line_17' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l18 including contents;
create tablespace ts_l18
datafile '/home/oracle/dev/raw/line_18' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l19 including contents;
create tablespace ts_l19
datafile '/home/oracle/dev/raw/line_19' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l20 including contents;
create tablespace ts_l20
datafile '/home/oracle/dev/raw/line_20' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l21 including contents;
create tablespace ts_l21
datafile '/home/oracle/dev/raw/line_21' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l22 including contents;
create tablespace ts_l22
datafile '/home/oracle/dev/raw/line_22' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l23 including contents;
create tablespace ts_l23
datafile '/home/oracle/dev/raw/line_23' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l24 including contents;
create tablespace ts_l24
datafile '/home/oracle/dev/raw/line_24' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*wait
```

```
*sql
{
--drop tablespace ts_l25 including contents;
create tablespace ts_l25
datafile '/home/oracle/dev/raw/line_25' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l26 including contents;
create tablespace ts_l26
datafile '/home/oracle/dev/raw/line_26' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l27 including contents;
create tablespace ts_l27
datafile '/home/oracle/dev/raw/line_27' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l28 including contents;
create tablespace ts_l28
datafile '/home/oracle/dev/raw/line_28' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l29 including contents;
create tablespace ts_l29
datafile '/home/oracle/dev/raw/line_29' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l30 including contents;
create tablespace ts_l30
datafile '/home/oracle/dev/raw/line_30' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l31 including contents;
create tablespace ts_l31
datafile '/home/oracle/dev/raw/line_31' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l32 including contents;
create tablespace ts_l32
datafile '/home/oracle/dev/raw/line_32' size 30720m reuse
extent management dictionary default storage (initial 320m next 10m maxextents unlimited pctincrease 0)
;
}
```

```
*wait
*sql
{
--drop tablespace ts_o1 including contents;
create tablespace ts_o1
datafile '/home/oracle/dev/raw/ord_1' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o2 including contents;
create tablespace ts_o2
datafile '/home/oracle/dev/raw/ord_2' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o3 including contents;
create tablespace ts_o3
datafile '/home/oracle/dev/raw/ord_3' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o4 including contents;
create tablespace ts_o4
datafile '/home/oracle/dev/raw/ord_4' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o5 including contents;
create tablespace ts_o5
datafile '/home/oracle/dev/raw/ord_5' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o6 including contents;
create tablespace ts_o6
datafile '/home/oracle/dev/raw/ord_6' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o7 including contents;
create tablespace ts_o7
datafile '/home/oracle/dev/raw/ord_7' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o8 including contents;
create tablespace ts_o8
datafile '/home/oracle/dev/raw/ord_8' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
```

```
}
*wait
*sql
{
--drop tablespace ts_o9 including contents;
create tablespace ts_o9
datafile '/home/oracle/dev/raw/ord_9' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o10 including contents;
create tablespace ts_o10
datafile '/home/oracle/dev/raw/ord_10' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o11 including contents;
create tablespace ts_o11
datafile '/home/oracle/dev/raw/ord_11' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o12 including contents;
create tablespace ts_o12
datafile '/home/oracle/dev/raw/ord_12' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o13 including contents;
create tablespace ts_o13
datafile '/home/oracle/dev/raw/ord_13' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o14 including contents;
create tablespace ts_o14
datafile '/home/oracle/dev/raw/ord_14' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o15 including contents;
create tablespace ts_o15
datafile '/home/oracle/dev/raw/ord_15' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o16 including contents;
create tablespace ts_o16
datafile '/home/oracle/dev/raw/ord_16' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
```

```
;
}
*wait
*sql
{
--drop tablespace ts_o17 including contents;
create tablespace ts_o17
datafile '/home/oracle/dev/raw/ord_17' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o18 including contents;
create tablespace ts_o18
datafile '/home/oracle/dev/raw/ord_18' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o19 including contents;
create tablespace ts_o19
datafile '/home/oracle/dev/raw/ord_19' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o20 including contents;
create tablespace ts_o20
datafile '/home/oracle/dev/raw/ord_20' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o21 including contents;
create tablespace ts_o21
datafile '/home/oracle/dev/raw/ord_21' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o22 including contents;
create tablespace ts_o22
datafile '/home/oracle/dev/raw/ord_22' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o23 including contents;
create tablespace ts_o23
datafile '/home/oracle/dev/raw/ord_23' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o24 including contents;
create tablespace ts_o24
datafile '/home/oracle/dev/raw/ord_24' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited
```

```
pctincrease 0)
;
}
*wait
*sql
{
--drop tablespace ts_o25 including contents;
create tablespace ts_o25
datafile '/home/oracle/dev/raw/ord_25' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o26 including contents;
create tablespace ts_o26
datafile '/home/oracle/dev/raw/ord_26' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o27 including contents;
create tablespace ts_o27
datafile '/home/oracle/dev/raw/ord_27' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o28 including contents;
create tablespace ts_o28
datafile '/home/oracle/dev/raw/ord_28' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o29 including contents;
create tablespace ts_o29
datafile '/home/oracle/dev/raw/ord_29' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o30 including contents;
create tablespace ts_o30
datafile '/home/oracle/dev/raw/ord_30' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o31 including contents;
create tablespace ts_o31
datafile '/home/oracle/dev/raw/ord_31' size 7680m reuse
extent management dictionary default storage (initial 75m next 5m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o32 including contents;
create tablespace ts_o32
datafile '/home/oracle/dev/raw/ord_32' size 7680m reuse
extent management dictionary default storage
```

```
(initial 75m next 5m maxextents unlimited
pctincrease 0)
;
}
*wait
*sql
{
--drop tablespace ts_rest including contents;
create tablespace ts_rest
datafile '/home/oracle/dev/raw/rest_1' size
7680m reuse
extent management dictionary default storage
(initial 1024m next 10m maxextents unlimited
pctincrease 0)
;
}
*sql
{
--drop tablespace ts_psupp including contents;
create tablespace ts_psupp
datafile '/home/oracle/dev/raw/psupp_1' size
6144m reuse
extent management dictionary default storage
(initial 3500m next 50m maxextents unlimited
pctincrease 0)
;
}
*sql
{
--drop tablespace ts_index including contents;
create tablespace ts_index
datafile '/home/oracle/dev/raw/index_1' size
9600m reuse
extent management local
uniform size 60M
;
}
# creating tpch's ts_temp tablespace
*wait
# adding tablespace datafiles
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_2' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_3' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_4' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_5' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_6' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_7' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_8' size
7680m reuse
;
```

```
}
*wait
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_9' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_10' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_11' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_12' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_13' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_14' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_15' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_16' size
7680m reuse
;
}
*wait
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_17' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_18' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_19' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_20' size
7680m reuse
;
}
*sql
```

```
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_21' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_22' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_23' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_24' size
7680m reuse
;
}
*wait
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_25' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_26' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_27' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_28' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_29' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_30' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_31' size
7680m reuse
;
}
*sql
{
alter tablespace ts_rest
add datafile '/home/oracle/dev/raw/rest_32' size
7680m reuse
;
}
*wait
*sql
{
alter tablespace ts_psupp
```

```
add datafile '/home/oracle/dev/raw/psupp_2' size
6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_3' size
6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_4' size
6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_5' size
6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_6' size
6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_7' size
6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_8' size
6144m reuse
;
}
*wait
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_9' size
6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_10'
size 6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_11'
size 6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_12'
size 6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_13'
size 6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_14'
size 6144m reuse
;
```

```
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_15'
size 6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_16'
size 6144m reuse
;
}
*wait
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_17'
size 6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_18'
size 6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_19'
size 6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_20'
size 6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_21'
size 6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_22'
size 6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_23'
size 6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_24'
size 6144m reuse
;
}
*wait
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_25'
size 6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_26'
size 6144m reuse
;
}
*sql
```

```
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_27'
size 6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_28'
size 6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_29'
size 6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_30'
size 6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_31'
size 6144m reuse
;
}
*sql
{
alter tablespace ts_psupp
add datafile '/home/oracle/dev/raw/psupp_32'
size 6144m reuse
;
}
*wait
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_2' size
9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_3' size
9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_4' size
9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_5' size
9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_6' size
9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_7' size
9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_8' size
```

```
9600m reuse
;
}
*wait
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_9' size
9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_10'
size 9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_11'
size 9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_12'
size 9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_13'
size 9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_14'
size 9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_15'
size 9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_16'
size 9600m reuse
;
}
*wait
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_17'
size 9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_18'
size 9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_19'
size 9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_20'
size 9600m reuse
;
```

```
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_21'
size 9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_22'
size 9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_23'
size 9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_24'
size 9600m reuse
;
}
*wait
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_25'
size 9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_26'
size 9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_27'
size 9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_28'
size 9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_29'
size 9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_30'
size 9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_31'
size 9600m reuse
;
}
*sql
{
alter tablespace ts_index
add datafile '/home/oracle/dev/raw/index_32'
size 9600m reuse
;
}
*wait
#adding tpch's ts_temp add datafiles
```

```
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_2' size
22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_3' size
22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_4' size
22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_5' size
22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_6' size
22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_7' size
22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_8' size
22272m reuse
;
}
*wait
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_9' size
22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_10'
size 22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_11'
size 22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_12'
size 22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_13'
size 22272m reuse
;
}
*sql
{
alter tablespace ts_temp
```

```
add tempfile '/home/oracle/dev/raw/temp_14'
size 22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_15'
size 22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_16'
size 22272m reuse
;
}
*wait
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_17'
size 22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_18'
size 22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_19'
size 22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_20'
size 22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_21'
size 22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_22'
size 22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_23'
size 22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_24'
size 22272m reuse
;
}
*wait
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_25'
size 22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_26'
size 22272m reuse
```

```
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_27'
size 22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_28'
size 22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_29'
size 22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_30'
size 22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_31'
size 22272m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/temp_32'
size 22272m reuse
;
}
*wait
*bgoff
%e-sctso
%b-dapop
*bgon=1
########################################
########################################
# Schema Creation Phase - User and Tables
# AND Database Population Phase
#
# creating tpch user
*sql
{
-- drop user tpch cascade;
grant DBA
to tpch identified by tpch;
}
*wait
*sql
{
connect tpch/tpch;
drop directory data_dir;
create directory data_dir as
'/oracle.data1/flatfiles';
}
*sql
{
connect tpch/tpch;
drop table l_et;
create table l_et(
    l_orderkey      number ,
    l_partkey       number ,
    l_suppkey       number ,
    l_linenumber    number ,
    l_quantity      number ,
    l_extendedprice number ,
    l_discount      number ,
    l_tax           number ,
    l_returnflag    char(1) ,
    l_linestatus    char(1) ,
    l_shipdate      date ,
    l_commitdate    date ,
    l_receiptdate   date ,
    l_shipinstruct  char(25) ,
```

```
    l_shipmode          char(10) ,
    l_comment           varchar(44)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
        records delimited by newline
        nobadfile
        nologfile
        fields terminated by '|'
        missing field values are null
    )
    location (
'L1','L2','L3','L4','L5','L6',
'L7','L8','L9','L10','L11','L12',
'L13','L14','L15','L16','L17','L18',
'L19','L20','L21','L22','L23','L24',
'L25','L26','L27','L28','L29','L30',
'L31','L32'
))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table o_et;
create table o_et(
    o_orderkey          number ,
    o_custkey           number ,
    o_orderstatus       char(1) ,
    o_totalprice        number ,
    o_orderdate         date ,
    o_orderpriority     char(15) ,
    o_clerk             char(15) ,
    o_shippriority      number ,
    o_comment           varchar(79)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
        records delimited by newline
        nobadfile
        nologfile
        fields terminated by '|'
        missing field values are null
    )
    location (
'O1','O2','O3','O4','O5','O6',
'O7','O8','O9','O10','O11','O12',
'O13','O14','O15','O16','O17','O18',
'O19','O20','O21','O22','O23','O24',
'O25','O26','O27','O28','O29','O30',
'O31','O32'
))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table ps_et;
create table ps_et(
    ps_partkey          number ,
    ps_suppkey          number ,
    ps_availqty         number ,
    ps_supplycost       number ,
    ps_comment          varchar(199)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
        records delimited by newline
        nobadfile
        nologfile
        fields terminated by '|'
        missing field values are null
    )
    location (
'PS1','PS2','PS3','PS4','PS5','PS6',
'PS7','PS8','PS9','PS10','PS11','PS12',
'PS13','PS14','PS15','PS16','PS17','PS18',
'PS19','PS20','PS21','PS22','PS23','PS24',
'PS25','PS26','PS27','PS28','PS29','PS30',
```

```
'PS31','PS32'
))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table p_et;
create table p_et(
      p_partkey              number ,
      p_name                 varchar(55) ,
      p_mfgr                 char(25) ,
      p_brand                char(10) ,
      p_type                 varchar(25) ,
      p_size                 number ,
      p_container            char(10) ,
      p_retailprice          number ,
      p_comment              varchar(23)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
            records delimited by newline
            nobadfile
            nologfile
            fields terminated by '|'
            missing field values are null
       )
       location (
'P1','P2','P3','P4','P5','P6',
'P7','P8','P9','P10','P11','P12',
'P13','P14','P15','P16','P17','P18',
'P19','P20','P21','P22','P23','P24',
'P25','P26','P27','P28','P29','P30',
'P31','P32'
))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table c_et;
create table c_et(
      c_custkey              number ,
      c_name                 varchar(25) ,
      c_address              varchar(40) ,
      c_nationkey            number ,
      c_phone                char(15) ,
      c_acctbal              number ,
      c_mktsegment           char(10) ,
      c_comment              varchar(117)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
            records delimited by newline
            nobadfile
            nologfile
            fields terminated by '|'
            missing field values are null
       )
       location (
'C1','C2','C3','C4','C5','C6',
'C7','C8','C9','C10','C11','C12',
'C13','C14','C15','C16','C17','C18',
'C19','C20','C21','C22','C23','C24',
'C25','C26','C27','C28','C29','C30',
'C31','C32'
))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table s_et;
create table s_et(
      s_suppkey              number ,
      s_name                 char(25) ,
      s_address              varchar(40) ,
      s_nationkey            number ,
      s_phone                char(15) ,
      s_acctbal              number ,
      s_comment              varchar(101)
)
```

```
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
        records delimited by newline
        nobadfile
        nologfile
        fields terminated by '|'
        missing field values are null
    )
    location (
'S1','S2','S3','S4','S5','S6',
'S7','S8','S9','S10','S11','S12',
'S13','S14','S15','S16','S17','S18',
'S19','S20','S21','S22','S23','S24',
'S25','S26','S27','S28','S29','S30',
'S31','S32'
    ))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table n_et;
create table n_et(
    n_nationkey         number ,
    n_name              char(25) ,
    n_regionkey         number ,
    n_comment           varchar(152)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
        records delimited by newline
        nobadfile
        nologfile
        fields terminated by '|'
        missing field values are null
    )
    location (
    'N1'))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table r_et;
create table r_et(
    r_regionkey         number ,
    r_name              char(25) ,
    r_comment           varchar(152)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
        records delimited by newline
        nobadfile
        nologfile
        fields terminated by '|'
        missing field values are null
    )
    location (
    'R1'))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
alter table l_et parallel;
alter table o_et parallel;
alter table ps_et parallel;
alter table p_et parallel;
alter table c_et parallel;
alter table s_et parallel;
}
# altering tpch's default and temporary tablespace
*sql
{
alter user tpch default tablespace ts_default;
alter user tpch temporary tablespace ts_temp;
}
*sql
```

```
{
connect tpch/tpch
@?/rdbms/admin/utlxplan.sql;
}
*wait
*sql
{
set timing on
set echo on
!date
connect tpch/tpch;
rem drop table lineitem;
create table lineitem(
    l_shipdate      ,
    l_orderkey      NOT NULL,
    l_discount      NOT NULL,
    l_extendedprice NOT NULL,
    l_suppkey       NOT NULL,
    l_quantity      NOT NULL,
    l_returnflag    ,
    l_partkey       NOT NULL,
    l_linestatus    ,
    l_tax           NOT NULL,
    l_commitdate    ,
    l_receiptdate   ,
    l_shipmode      ,
    l_linenumber    NOT NULL,
    l_shipinstruct  ,
    l_comment
)
pctfree 1
pctused 99
initrans 10
storage (initial 320m next 10m freelist groups 4
freelists 99)
parallel
nologging
partition by range (l_shipdate)
subpartition by hash(l_partkey)
subpartitions 32
(
partition item1 values less than
(to_date('1992-01-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item2 values less than
(to_date('1992-02-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item3 values less than
(to_date('1992-03-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item4 values less than
(to_date('1992-04-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item5 values less than
(to_date('1992-05-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item6 values less than
(to_date('1992-06-01','YYYY-MM-DD'))
```

```
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item7 values less than
(to_date('1992-07-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item8 values less than
(to_date('1992-08-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item9 values less than
(to_date('1992-09-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item10 values less than
(to_date('1992-10-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item11 values less than
(to_date('1992-11-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item12 values less than
(to_date('1992-12-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item13 values less than
(to_date('1993-01-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item14 values less than
(to_date('1993-02-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item15 values less than
(to_date('1993-03-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
```

```
partition item16 values less than
(to_date('1993-04-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item17 values less than
(to_date('1993-05-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item18 values less than
(to_date('1993-06-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item19 values less than
(to_date('1993-07-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item20 values less than
(to_date('1993-08-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item21 values less than
(to_date('1993-09-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item22 values less than
(to_date('1993-10-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item23 values less than
(to_date('1993-11-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item24 values less than
(to_date('1993-12-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item25 values less than
(to_date('1994-01-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
```

```
ts_l32)
,
partition item26 values less than (to_date('1994-02-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6, ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13, ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19, ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25, ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31, ts_l32)
,
partition item27 values less than (to_date('1994-03-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6, ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13, ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19, ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25, ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31, ts_l32)
,
partition item28 values less than (to_date('1994-04-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6, ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13, ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19, ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25, ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31, ts_l32)
,
partition item29 values less than (to_date('1994-05-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6, ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13, ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19, ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25, ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31, ts_l32)
,
partition item30 values less than (to_date('1994-06-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6, ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13, ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19, ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25, ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31, ts_l32)
,
partition item31 values less than (to_date('1994-07-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6, ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13, ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19, ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25, ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31, ts_l32)
,
partition item32 values less than (to_date('1994-08-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6, ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13, ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19, ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25, ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31, ts_l32)
,
partition item33 values less than (to_date('1994-09-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6, ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13, ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19, ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25, ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31, ts_l32)
,
partition item34 values less than (to_date('1994-10-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6, ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13, ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19, ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25, ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31, ts_l32)
,
partition item35 values less than (to_date('1994-11-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6, ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13, ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
```

```
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item36 values less than
(to_date('1994-12-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item37 values less than
(to_date('1995-01-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item38 values less than
(to_date('1995-02-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item39 values less than
(to_date('1995-03-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item40 values less than
(to_date('1995-04-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item41 values less than
(to_date('1995-05-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item42 values less than
(to_date('1995-06-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item43 values less than
(to_date('1995-07-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item44 values less than
(to_date('1995-08-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item45 values less than
(to_date('1995-09-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
```

```
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item46 values less than
(to_date('1995-10-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item47 values less than
(to_date('1995-11-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item48 values less than
(to_date('1995-12-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item49 values less than
(to_date('1996-01-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item50 values less than
(to_date('1996-02-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item51 values less than
(to_date('1996-03-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item52 values less than
(to_date('1996-04-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item53 values less than
(to_date('1996-05-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item54 values less than
(to_date('1996-06-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item55 values less than
```

```
(to_date('1996-07-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item56 values less than
(to_date('1996-08-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item57 values less than
(to_date('1996-09-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item58 values less than
(to_date('1996-10-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item59 values less than
(to_date('1996-11-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item60 values less than
(to_date('1996-12-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item61 values less than
(to_date('1997-01-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item62 values less than
(to_date('1997-02-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item63 values less than
(to_date('1997-03-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item64 values less than
(to_date('1997-04-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
```

```
,
partition item65 values less than
(to_date('1997-05-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item66 values less than
(to_date('1997-06-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item67 values less than
(to_date('1997-07-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item68 values less than
(to_date('1997-08-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item69 values less than
(to_date('1997-09-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item70 values less than
(to_date('1997-10-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item71 values less than
(to_date('1997-11-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item72 values less than
(to_date('1997-12-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item73 values less than
(to_date('1998-01-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item74 values less than
(to_date('1998-02-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
```

```
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item75 values less than
(to_date('1998-03-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item76 values less than
(to_date('1998-04-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item77 values less than
(to_date('1998-05-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item78 values less than
(to_date('1998-06-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item79 values less than
(to_date('1998-07-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item80 values less than
(to_date('1998-08-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item81 values less than
(to_date('1998-09-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item82 values less than
(to_date('1998-10-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item83 values less than
(to_date('1998-11-01','YYYY-MM-DD'))
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32)
,
partition item84 values less than (MAXVALUE)
store in (ts_l1, ts_l2, ts_l3, ts_l4, ts_l5, ts_l6,
ts_l7, ts_l8, ts_l9, ts_l10, ts_l11, ts_l12, ts_l13,
ts_l14, ts_l15, ts_l16, ts_l17, ts_l18, ts_l19,
```

```
ts_l20, ts_l21, ts_l22, ts_l23, ts_l24, ts_l25,
ts_l26, ts_l27, ts_l28, ts_l29, ts_l30, ts_l31,
ts_l32))
as select
    l_shipdate      ,
    l_orderkey      ,
    l_discount      ,
    l_extendedprice ,
    l_suppkey       ,
    l_quantity      ,
    l_returnflag    ,
    l_partkey       ,
    l_linestatus    ,
    l_tax           ,
    l_commitdate    ,
    l_receiptdate   ,
    l_shipmode      ,
    l_linenumber    ,
    l_shipinstruct  ,
    l_comment
from l_et;
!date
}
*wait
*sql
{
connect tpch/tpch;
set timing on
set echo on
!date

rem drop table orders;
create table orders(
    o_orderdate     ,
    o_orderkey      NOT NULL,
    o_custkey       NOT NULL,
    o_orderpriority ,
    o_shippriority  ,
    o_clerk         ,
    o_orderstatus   ,
    o_totalprice    ,
    o_comment
)
pctfree 1
pctused 99
initrans 10
storage (initial 75m next 5m freelist groups 4
freelists 99)
parallel
nologging
partition by range (o_orderdate)
subpartition by hash(o_custkey)
subpartitions 32
(
partition ord1 values less than
(to_date('1992-01-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord2 values less than
(to_date('1992-02-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord3 values less than
(to_date('1992-03-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord4 values less than
(to_date('1992-04-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
```

```
ts_o31, ts_o32)
,
partition ord5 values less than (to_date('1992-05-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18, ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24, ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30, ts_o31, ts_o32)
,
partition ord6 values less than (to_date('1992-06-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18, ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24, ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30, ts_o31, ts_o32)
,
partition ord7 values less than (to_date('1992-07-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18, ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24, ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30, ts_o31, ts_o32)
,
partition ord8 values less than (to_date('1992-08-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18, ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24, ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30, ts_o31, ts_o32)
,
partition ord9 values less than (to_date('1992-09-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18, ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24, ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30, ts_o31, ts_o32)
,
partition ord10 values less than (to_date('1992-10-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18, ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24, ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30, ts_o31, ts_o32)
,
partition ord11 values less than (to_date('1992-11-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18, ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24, ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30, ts_o31, ts_o32)
,
partition ord12 values less than (to_date('1992-12-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18, ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24, ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30, ts_o31, ts_o32)
,
partition ord13 values less than (to_date('1993-01-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18, ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24, ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30, ts_o31, ts_o32)
,
partition ord14 values less than (to_date('1993-02-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
```

```
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord15 values less than
(to_date('1993-03-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord16 values less than
(to_date('1993-04-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord17 values less than
(to_date('1993-05-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord18 values less than
(to_date('1993-06-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord19 values less than
(to_date('1993-07-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord20 values less than
(to_date('1993-08-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord21 values less than
(to_date('1993-09-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord22 values less than
(to_date('1993-10-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord23 values less than
(to_date('1993-11-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord24 values less than
(to_date('1993-12-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
```

```
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord25 values less than
(to_date('1994-01-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord26 values less than
(to_date('1994-02-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord27 values less than
(to_date('1994-03-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord28 values less than
(to_date('1994-04-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord29 values less than
(to_date('1994-05-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord30 values less than
(to_date('1994-06-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord31 values less than
(to_date('1994-07-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord32 values less than
(to_date('1994-08-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord33 values less than
(to_date('1994-09-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord34 values less than
```

```
(to_date('1994-10-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord35 values less than
(to_date('1994-11-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord36 values less than
(to_date('1994-12-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord37 values less than
(to_date('1995-01-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord38 values less than
(to_date('1995-02-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord39 values less than
(to_date('1995-03-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord40 values less than
(to_date('1995-04-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord41 values less than
(to_date('1995-05-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord42 values less than
(to_date('1995-06-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord43 values less than
(to_date('1995-07-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
```

```
,
partition ord44 values less than (to_date('1995-08-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18, ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24, ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30, ts_o31, ts_o32)
,
partition ord45 values less than (to_date('1995-09-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18, ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24, ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30, ts_o31, ts_o32)
,
partition ord46 values less than (to_date('1995-10-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18, ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24, ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30, ts_o31, ts_o32)
,
partition ord47 values less than (to_date('1995-11-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18, ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24, ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30, ts_o31, ts_o32)
,
partition ord48 values less than (to_date('1995-12-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18, ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24, ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30, ts_o31, ts_o32)
,
partition ord49 values less than (to_date('1996-01-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18, ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24, ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30, ts_o31, ts_o32)
,
partition ord50 values less than (to_date('1996-02-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18, ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24, ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30, ts_o31, ts_o32)
,
partition ord51 values less than (to_date('1996-03-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18, ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24, ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30, ts_o31, ts_o32)
,
partition ord52 values less than (to_date('1996-04-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18, ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24, ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30, ts_o31, ts_o32)
,
partition ord53 values less than (to_date('1996-05-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6, ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12, ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18, ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
```

```
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord54 values less than
(to_date('1996-06-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord55 values less than
(to_date('1996-07-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord56 values less than
(to_date('1996-08-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord57 values less than
(to_date('1996-09-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord58 values less than
(to_date('1996-10-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord59 values less than
(to_date('1996-11-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord60 values less than
(to_date('1996-12-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord61 values less than
(to_date('1997-01-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord62 values less than
(to_date('1997-02-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord63 values less than
(to_date('1997-03-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
```

```
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord64 values less than
(to_date('1997-04-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord65 values less than
(to_date('1997-05-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord66 values less than
(to_date('1997-06-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord67 values less than
(to_date('1997-07-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord68 values less than
(to_date('1997-08-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord69 values less than
(to_date('1997-09-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord70 values less than
(to_date('1997-10-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord71 values less than
(to_date('1997-11-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord72 values less than
(to_date('1997-12-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord73 values less than
(to_date('1998-01-01','YYYY-MM-DD'))
```

```
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord74 values less than
(to_date('1998-02-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord75 values less than
(to_date('1998-03-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord76 values less than
(to_date('1998-04-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord77 values less than
(to_date('1998-05-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord78 values less than
(to_date('1998-06-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord79 values less than
(to_date('1998-07-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord80 values less than
(to_date('1998-08-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord81 values less than
(to_date('1998-09-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord82 values less than
(to_date('1998-10-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
```

```
partition ord83 values less than
(to_date('1998-11-01','YYYY-MM-DD'))
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
,
partition ord84 values less than (MAXVALUE)
store in (ts_o1, ts_o2, ts_o3, ts_o4, ts_o5, ts_o6,
ts_o7, ts_o8, ts_o9, ts_o10, ts_o11, ts_o12,
ts_o13, ts_o14, ts_o15, ts_o16, ts_o17, ts_o18,
ts_o19, ts_o20, ts_o21, ts_o22, ts_o23, ts_o24,
ts_o25, ts_o26, ts_o27, ts_o28, ts_o29, ts_o30,
ts_o31, ts_o32)
)
as select
    o_orderdate     ,
    o_orderkey      ,
    o_custkey       ,
    o_orderpriority ,
    o_shippriority  ,
    o_clerk         ,
    o_orderstatus   ,
    o_totalprice    ,
    o_comment
from o_et;
!date
}
*wait
*sql
{
connect tpch/tpch
set timing on
set echo on

!date

rem drop table partsupp;
create table partsupp(
    ps_partkey      NOT NULL,
    ps_suppkey      NOT NULL,
    ps_supplycost   NOT NULL,
    ps_availqty     ,
    ps_comment      ,
constraint pk_partkey_suppkey_1 primary
key(ps_partkey, ps_suppkey)
)
organization index
partition by hash(ps_partkey)
partitions 32
storage (initial 1024m next 50m)
parallel
nologging
pctthreshold 50
tablespace ts_psupp
as select
    ps_partkey      ,
    ps_suppkey      ,
    ps_supplycost   ,
    ps_availqty     ,
    ps_comment
from ps_et;
!date
}
*wait
*sql
{
connect tpch/tpch
set timing on
set echo on

!date
rem drop table customer;
create table customer(
    c_custkey       NOT NULL,
    c_mktsegment    ,
    c_nationkey     ,
    c_name          ,
    c_address       ,
    c_phone         ,
    c_acctbal       ,
    c_comment
)
pctfree 0
pctused 99
parallel
```

```
nologging
storage (initial 850m next 10m)
partition by hash (c_custkey)
partitions 32
tablespace ts_rest
as select
    c_custkey       ,
    c_mktsegment    ,
    c_nationkey     ,
    c_name          ,
    c_address       ,
    c_phone         ,
    c_acctbal       ,
    c_comment
from c_et;
!date
}
*wait
*sql
{
connect tpch/tpch
set timing on
set echo on

!date
rem drop table part;

create table part(
    p_partkey       NOT NULL,
    p_type          ,
    p_size          ,
    p_brand         ,
    p_name          ,
    p_container     ,
    p_mfgr          ,
    p_retailprice   ,
    p_comment
)
pctfree 0
pctused 99
parallel
nologging
storage (initial 1024m next 20m)
partition by hash (p_partkey)
partitions 32
tablespace ts_rest
as select
    p_partkey       ,
    p_type          ,
    p_size          ,
    p_brand         ,
    p_name          ,
    p_container     ,
    p_mfgr          ,
    p_retailprice   ,
    p_comment
from p_et;
!date
}
*wait
*sql
{
connect tpch/tpch;
set timing on
set echo on
rem drop table supplier;
create table supplier(
    s_suppkey       NOT NULL,
    s_nationkey     ,
    s_comment       ,
    s_name          ,
    s_address       ,
    s_phone         ,
    s_acctbal
)
pctfree 0
pctused 99
parallel
nologging
storage (initial 75m next 5m)
partition by hash (s_suppkey)
partitions 32
tablespace ts_rest
as select
    s_suppkey       ,
    s_nationkey     ,
    s_comment       ,
    s_name          ,
```

```
    s_address       ,
    s_phone         ,
    s_acctbal
from s_et;
}
*wait
*sql
{
connect tpch/tpch;
set echo on
set timing on

rem drop table nation;
create table nation(
    n_nationkey     NOT NULL,
    n_name          ,
    n_regionkey     ,
    n_comment       )
tablespace ts_default
as select * from n_et;

rem drop table region;
create table region(
    r_regionkey     ,
    r_name          ,
    r_comment       )
tablespace ts_default
as select * from r_et;
}

*wait
*bgoff
%e-scuto

*sql
{
connect tpch/tpch
set timing on
set echo on

!date
drop table l_et;
drop table o_et;
drop table ps_et;
drop table p_et;
drop table c_et;
drop table s_et;
drop table n_et;
drop table r_et;
}
*bgoff
%e-dapop
%b-ixcre
*bgon=1
########################################
########################################
# Index Creation Phase
*sql
{
connect tpch/tpch;
!date
set echo on
set timing on
rem drop index i_l_orderkey;
create index i_l_orderkey
on lineitem (l_orderkey) global partition by hash
(l_orderkey)
partitions 32
pctfree 5
initrans 10
tablespace ts_index
storage (freelist groups 4 freelists 99)
parallel
compute statistics
nologging;
}
*sql
{
connect tpch/tpch;
!date
set echo on
set timing on
rem drop index i_o_orderkey;
create unique index i_o_orderkey
on orders (o_orderkey) global partition by hash
(o_orderkey)
partitions 32
pctfree 5
```

```
initrans 10
tablespace ts_index
storage (freelist groups 4 freelis ts 99 )
parallel
compute statistics
nologging;
}
*sql
{
connect tpch/tpch;
!date
set echo on
set timing on

rem drop index i_c_custkey;
create unique index i_c_custkey
on customer (c_custkey)
pctfree 5
initrans 10
tablespace ts_index
storage (freelist groups 4 freelists 99)
parallel
compute statistics
nologging;
}
*wait

#!/usr/bin/perl
#
# $Header: bumpxlite.pl 23-oct-2002.13:15:45 mpoess Exp $
#
# bumpxlite.pl
#
# Copyright (c) 2001, 2002, Oracle Corporation. All rights reserved.
#
#      NAME
#        bumpxlite.pl - <one-line expansion of the name>
#
#      DESCRIPTION
#          <short description of component this file declares/defines>
*bgoff
%e-ixcre
%b-anlyz
*bgon=1
########################################
########################################
# Analyze Phase
*sql
{
connect tpch/tpch;
!date
set timing on
execute
dbms_stats.gather_schema_stats('TPCH' ,
estimate_percent => 1, degree => 32 ,
granularity => 'GLOBAL' );
connect / as sysdba
execute dbms_stats.gather_system_stats;
alter system switch logfile;
!date
}
*wait
*bgoff
%e-anlyz
```

B.2 bumpx.pl

```
#
#      NOTES
#        <other useful comments, qualifications, etc.>
#
#      MODIFIED     (MM/DD/YY)
#               mpoess                10/23/02 - mpoess_update_from_visa
#      mpoess          09/24/01  - take out readfile subroutine
#      mpoess          08/10/01 - Creation
#

$os = $ENV{'OS'};
if (($os cmp 'Windows_NT') != 0) { # os is UNIX
    $os = "unix";$nt = 0;$unix = 1;
} else {
```

```
    $os = "nt"; $nt = 1; $unix = 0;
}
$| = 1;
$verbose = 0;
if (($os cmp "unix")==0) {
      $defphases                                    =
"dbcre,sctso,scuto,dbgen,dapop,anlyz,ixcre";
} else {
      $defphases                                    =
"sdgen,shutd,start,dbgen,plcre,dbcre,sctso,scuto,
dapop,scuvo,anlyz,ixcre,chob";
}
$allbmtypes = "tpcd,wisc";
$bmtype = "tpcd" if !defined $bmtype;
$pdfile =  "$ENV{'BUMPX_DIR'}/param.txt";
# This file contains the description of all possible
parameters.
while ($arg = shift(@ARGV)) {
    if ($arg !~ /-(i|o|t|p|d|a|s|h)/){
         $error =  "** Error: Bad argument to
$0: $arg\n";
          &usage;
     }
    if ($arg =~ /-h/) { &usage;ext(0);}
     $runsilent = 1 if ($arg =~ /-s/);
    $outfile = shift(@ARGV) if ($arg =~ /-o/);
    $bmtype = shift(@ARGV) if ($arg =~ /-t/);
      $phaselist = shift(@ARGV) if ($arg =~
/-p/);
    if ($arg =~ /-d/) {
      $defpar = shift(@ARGV);
      @keys = keys %params;
      while ($#keys >=0) {
          $key = pop(@keys);
         if (($defpar cmp "") ==0) {
           print $key, "=", $params{$key}, "\n";
          } else {
           print $key, "=", $params{$key}, "\n"
if ($key =~ /$defpar/);
          }
      }
      exit(0);
     }
}
$outfile = "$ENV{'BUMPX_DIR'}/bumpx.dat"
if !defined $outfile;
if ($nt) {
     $listdir = $filedir."list/";
     if (!-e $listfile) {
       system ("mkdir $listdir");
     }
}
if (($os cmp "nt") == 0) {  ##  NT  Port  (Use
tmpfile to buffer
     $tmpfile = "tmp.txt";          ## commands
and nruntpb to synchronize them)
     $tmpfile = $filedir.$tmpfile;
     $nruntpb =  "nruntpb.exe";
} ## NT End
if (!-e $outfile) {
    $error = "** Error: -o file, $outfile, does not
exist\n";
     &usage;
}
$phaselist = $defphases if !defined $phaselist;
@phases = split(/,/, $phaselist);
## NT Port (Use tmpfile to buffer commands for
nruntpb)
open (TMPFILE, ">$tmpfile") if ( (($os cmp
"nt") == 0));
## NT End
&doexecute;
## NT Port
close(TMPFILE) if ( (($os cmp "nt") == 0));
## NT End
exit(0);

sub doexecute { # First, do preprocessing stuff
    print "Execution pass begun." if $verbose;
     open (INFILE, $outfile);
   WLOOP1:
     while ($line = <INFILE>)
      {
          study $line;
        next WLOOP1 if $line =~ /^\s*\#/;
        next WLOOP1 if $line =~ /^\s*\n/;
          if ($line =~ /^%b-preproc/)
           {
```

```
            $insection = 1;
             next WLOOP1;
          }
        next WLOOP1 if ($insection != 1);
         if ($line =~ /^%e-preproc/)
          {
            $insection = 0;
              $commands{$shortcmd}          =
$longcmd if defined $shortcmd;
             last WLOOP1;
          }
        if ($line =~ /^\*/)
          {
              $commands{$shortcmd}          =
$longcmd if defined $shortcmd;
            $line =~ /^(\*.*\S+)\s*\n$/;
            $shortcmd = $1;
            $longcmd = "";
             next WLOOP1;
          }
        if ($line =~ /^\\/)
          {
            #$line =~ /\\(.*\n)/;
            $line =~ /^\\(.*\n)/;
           $longcmd = $longcmd . $1;
             next WLOOP1;
          }
         print "Illegal entry in preproc stage:\n
$line";
      }
     close (INFILE);

# Then, do all of the requested phases
     $execctr = 0;
     foreach $phase (@phases)
      {
       $phase_cmd_num = 0;
       print '\n   Executing phase \'$phase\'" if
$verbose;
       $bg = 0;
          open (INFILE, $outfile);
        WLOOP2:
         while ($line = <INFILE>)
          {
           study $line;
             next    WLOOP2    if    $line    =~
/^\s*\#/;
             next    WLOOP2    if    $line    =~
/^\s*\n/;
          if ($line =~ /^\*ignon/)
           {
             $ignon = 1;
              next WLOOP2;
           }
          if ($line =~ /^\*ignoff/)
           {
             $ignon = 0;
              next WLOOP2;
           }
         next WLOOP2 if ($ignon == 1);
          if ($line =~ /^%b-$phase/)
           {
             $insection = 1;
             $execcmd = "";
              next WLOOP2;
           }
            next WLOOP2 if ($insection !=
1);
          if ($line =~ /^%e-$phase/)
           {
              $insection = 0;
              &execute ($execcmd);
              last WLOOP2;
           }
          if ($line =~ /^\*(.*)/)
           {
              &execute ($execcmd);
                if (($1 =~ /bgo/) || ($1 =~
/wait/) || ($1 =~ /ignore/))
                 {
                   $execcmd = $line;
                    next WLOOP2;
                 }
                $line =~ /^(\*.*\S+)\s*\n$/;
                  $execcmd                    =
$commands{$1};
                next WLOOP2;
           }
```

```
            if ($line =~ /^\{(.*)\}/)
            {
                $insert = "";
                $insert = $1;
                $execcmd =~ s/\{\}/$insert/;
                next WLOOP2;
            }
           if ($line =~ /^\{(.*)$/)
            {
                $insubsection = 1;
                $insert = "";
                $insert = $1;
                next WLOOP2;
            }
           if ($line =~ /^(.*)\}/)
            {
                $insubsection = 0;
                 $insert = $insert . $1;
          if (($os cmp "nt") == 0){ ## NT Port
(Ignore '\n')
              $insert =~ /(.*)\n$/s;
              $insert = $1;
          } ## NT End
                $execcmd =~ s/\{\}/$insert/;
                next WLOOP2;
            }
              $insert = $insert . $line if
($insubsection == 1);
        }
       close (INFILE);
    }
     print "\nExecution pass complete.\n" if
$verbose;
}

sub execute
{
   $cmd = shift(@_);
   if ($cmd)
   {
      return if ($cmd =~ /^\*ignore/);
       if ($cmd =~ /^\*bgon=(.*)/)
        {
            $bgmax = $1;
            $bg = 1;
            $bgrun = 0;
             return;
        }
      if ($cmd =~ /^\*bgoff/)
       {
            $bg = 0;
             return;
       }

    if ($cmd =~ /^\*time=(.*)/)    ##NT only
       {
       print $1 . "\n";
       print localtime(time) . "\n";
            return;
       }
      if ($cmd =~ /^copy (.*)/) ## NT only
       {
        system($cmd);
          ## Quit if failed
          if ($?) {
         print "system copy command
failed:\n$cmd\nreason: $? ($!)\n";
            exit(-1);
        }
            return;
       }
    if ($cmd =~ /^del (.*)/) ## NT only
       {
        system($cmd);
          ## Quit if failed
          if ($?) {
         print "system del command
failed:\n$cmd\nreason: $? ($!)\n";
            exit(-1);
        }
            return;
       }

    if ($cmd =~ /^\*wait/) ## This deals with
main differences between NT and UNIX
       {
       if (($os cmp "unix") == 0)
        {
```

```
            while ($fpid = shift(@wpids))
            {
                waitpid($fpid, 0);
            }
           }
           else
           {       ## NT Port (Start background tasks if any. nruntpb will wait until all tasks are done)
            if ($bgrun >= 1)
            {
                close(TMPFILE);
                system("cat $tmpfile >> $listdir$phase.lst");
               system("vi $tmpfile") if $debug;
                system("$nruntpb -p < $tmpfile") if !$debug;
               if ($?)
               {
                print "system command failed:\n$nruntpb < $tmpfile\n";
                print "reason: $? ($!)\n";
                print "Please check the contents in the input file.\n";
                exit(-1);
               }
               open(TMPFILE, ">$tmpfile");
            }
           }
              $bgrun = 0;
               return;
          }
         if ($cmd =~ /(s|g)etenv/)
          {
          @lines = split(/\n/, $cmd);
              $cmd = "";
              foreach $line (@lines)
               {
                   while (1)
                   {
                        last if ($line !~ /getenv/);
                        $line =~ /(.*)\*getenv\(([^(\)|\*)]*)\)(.*)/;
                        $line = $1 . $ENV{$2} . $3;
                   }
                    if ($line =~ /jojo/) #we do not want to use this for now
                   {
                        $line =~ /setenv\s+(\S+)\s+(\S+)/;
                       $ENV{$1} = $2;
                   }
                   else
                   {
                        $cmd = $cmd . $line. "\n";
                   }
               }
           }
            return if ($cmd !~ /\S+/); # return if nothing left to execute
        $execctr++;
            $ENV{'BUMPX_CTR'} = $$.'-'.$execctr;
        if (($os cmp "unix") == 0)
        {
            if ($bg == 1)
            {
             print "." if $verbose;
             $fpid = fork;
             if ($fpid == 0)
             {
                 exec ($cmd);
                  print "exec\'d command failed:\n$cmd\nreason: $!\n";
                  exit(-1);
             }
             unshift (@wpids, $fpid);
             $bgrun = $bgrun + 1;
             &execute ("*wait") if (($bgrun >= $bgmax) && ($bgmax >= 0));
            }
            else
            {
             system ($cmd);
             print "system\'d command
```

```
failed:\n$cmd\nreason: $? ($!)\n" if $?;
            }
        }
        else ## NT support
        {
           ## NT Port (Submit background tasks if there are bgmax of them, otherwise write to tmpfile)
            if ($bg == 1)
            {
              print "." if $verbose;
              if ($bgrun < $bgmax)
              {
                    $cmd =~ s/phase\#.lst/$listdir$phase\_$phase_cmd_num.lst/;
                   ++$phase_cmd_num;
                   print TMPFILE $cmd;
                  $bgrun = $bgrun + 1;
              }
              else
              {
                   close(TMPFILE);
                   system("cat $tmpfile >> $listdir$phase.lst");
                   system("$nruntpb -p < $tmpfile");
                  if ($?) {
                   print "system command failed:\n$nruntpb < $tmpfile\nreason: $? ($!)\n";
                   print "Please check the contents in the input file.\n";
                   exit(-1);
                  }
                  open(TMPFILE, ">$tmpfile");
                   $cmd =~ s/phase\#.lst/$listdir$phase\_$phase_cmd_num.lst/;
                   ++$phase_cmd_num;
                  print TMPFILE $cmd;
                  $bgrun = 1;
              }
             }
             else
             {
              $cmd =~ s/phase\#.lst/$listdir$phase\_$phase_cmd_num.lst/;
              ++$phase_cmd_num;
              print TMPFILE $cmd;
              close(TMPFILE);
              system("cat $tmpfile >> $listdir$phase.lst");
              system ("sh $tmpfile");
              if ($?) {
                   print "system\'d command failed:\nsh $tmpfile\nreason: $? ($!)\n";
                   print "Please check the contents in the shell script.\n";
                   exit(-1);
              }
              open(TMPFILE, ">$tmpfile");
             }
        } ## NT support End
       }
}

sub usage
{
    print "Usage:\n";
   print "This is a lite version of bumpx.pl. It can only be used to execute a .dat file\n";
    print " $0 [-o outfile] [-p phaselist] [-t type]\n";
    print "    -o : intermediary file to be created and/or used\n";
    print "         defaults to bumpx.dat in \$BUMPX_DIR or \$CWD\n";
    print "    -p : list of phases to create/execute\n";
    print "         phaselist is a comma separated list of phases in order\n";
    print "       possible phases are:\n";
     print "           sdgen = seed file generation\n";
    print "           dbgen = data flat file generation\n";
    print "       plcre = NT raw partition
```

```
and links creation\n";
creation\n";
        print "                shutd = shutdown
database (on all instances)\n";
      print "             start = startup database
(on all instances)\n";
        print "                sccre = schema
creation\n";
      print "                sctso = schema
creation (tablespaces only)\n";
        print "                scuto = schema
creation (user and tables only)\n";
        print "                scuvo = schema
creation (views only)\n";
         print "                  dapop = data
population\n";
      print "             ixcre = index creation
(including constraints)\n";
        print "                anlyz = analyze
objects\n";
        print "                chob = change
parameters of objects\n";
      print "             expln = create explain
plans\n";
      print "             dbcre = database
       print "               query = run and time
queries\n";
      print "          defaults to $defphases\n";
      print "    -t : type of benchmark\n";
       print    "                                enables
benchmark-specific defaults\n";
       print "          current possibilties are:
$allbmtypes\n";
      print "          defaults to tpcd\n";
       print "    -s : run silent (no parameter
checking is done)\n";
      print "\n";
      print "Examples:\n";
      print " $0 -p dapop\n";
      print "   Executes data population phase
of intermediary file bumpx.dat.\n";
      print "\n";
      print "$error\n";
       exit(-1);
}
```

Appendix C ACID Scripts

c.1 a_query.sql

```
Rem
Rem $Header: a_query.sql 06-aug-99.10:51:10 mpoess Exp $
Rem
Rem a_query.sql
Rem
Rem  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
Rem
Rem      NAME
Rem        a_query.sql - <one-line expansion of the name>
Rem
rem   DESCRIPTION
Rem        Performs ACID Query for TPC-D benchmark.
Rem          Asks user to input values for o_key
Rem       The range of okey is 1 to 600000
Rem
================================================================
Rem
Rem Usage:     sqlplus tpcd/tpcd @a_query <o_key>
Rem
Rem
Rem      MODIFIED     (MM/DD/YY)
Rem      mpoess         08/06/99 - Creation
Rem      mpoess         08/06/99 - Created
Rem


set serverout on;

select
'BEFORE ACID QUERY' as STAGE,
substr(TO_CHAR(sysdate,'YYYY-MM-DD HH:MI:SS'),1,20) as CURRENT_TIME
from dual;

select    SUM(trunc(trunc(l_extendedprice    *    (1-l_discount),2) * (1+l_tax),2)) AS RESULT
from lineitem
where l_orderkey = &&1;

select
'AFTER ACID QUERY' as STAGE,
substr(TO_CHAR(sysdate,'YYYY-MM-DD HH:MI:SS'),1,20) as CURRENT_TIME
from dual;

exit;
```

c.2 a_query2.sql

```
Rem
Rem $Header: aquery2.sql 07-aug-99.23:54:47 mpoess Exp $
Rem
Rem aquery2.sql
Rem
Rem  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
Rem
Rem      NAME
Rem        aquery2.sql - <one-line expansion of the name>
Rem
Rem   DESCRIPTION
Rem        Performs query on PARTSUPP for TPC-D benchmark
```

```
Rem        Isolation Test 5.
Rem          Asks user to input values for
ps_partkey and ps_suppkey
Rem         The range for ps_partkey is 1 to
20000
Rem        The range for ps_suppkey is 1 to
1000
Rem       A valid combination is 46 and 47
Rem  Usage:   sqlplus tpcd/tpcd @a_query2
<ps_partkey> <ps_suppkey>
Rem
Rem     MODIFIED    (MM/DD/YY)
Rem     mpoess      08/07/99 - Creation
Rem     mpoess      08/07/99 - Created
Rem
rem  DESCRIPTION
rem        Performs query on PARTSUPP for
TPC-D benchmark
rem        Isolation Test 5.
rem          Asks user to input values for
ps_partkey and ps_suppkey
rem         The range for ps_partkey is 1 to
20000
rem        The range for ps_suppkey is 1 to
1000
rem       A valid combination is 46 and 47

set serverout on;

select
'BEFORE PARTSUPP QUERY' as STAGE,
substr(TO_CHAR(sysdate,'YYYY-MM-DD
HH:MI:SS'),1,20) as CURRENT_TIME
from dual;

select *
from partsupp
where ps_partkey = &&1
and ps_suppkey = &&2;

select
'AFTER PARTSUPP QUERY' as STAGE,
substr(TO_CHAR(sysdate,'YYYY-MM-DD
HH:MI:SS'),1,20) as CURRENT_TIME
from dual;

exit;
```

c.3 atom.sh

```
#!/bin/ksh
#
#   $Header:   atom.sh   08-aug-99.13:48:02
mpoess Exp $
#
# atom.sh
#
#  Copyright (c) Oracle Corporation 1999. All
Rights Reserved.
#
#      NAME
#        atom.sh - <one-line expansion of the
name>
#
#   DESCRIPTION
#        Performs atomicity tests.
#         Usage: atom.sh [-n iter] [-p prog] [-u
usr/pswd] -h
#
#       Options: See usage below
#
#      NOTES
#                  <other useful comments,
qualifications, etc.>
#
#      MODIFIED    (MM/DD/YY)
#      mpoess      08/08/99 - Creation
#      mpoess      08/08/99 - Creation
#

. $KIT_DIR/env

OH=$ORACLE_HOME
#  ACID_DIR=$TPCD_KIT_DIR/audit set in
```

```
env
OUT_DIR=$ACID_OUT
DURA_DIR=$ACID_DIR/dura

usage() {

   echo ""
    echo "Usage: $0 [-n iter] [-p prog] [-u usr/pswd] -h"
   echo ""
    echo "-n iter       : number of iterations, default is 100"
    echo "-p prog       : program to run, default is atranspl.ott"
   echo "-u usr/pswd : user/password combo for database access, default is tpcd/tpcd"
    echo "-h            : print this usage summary"
   exit 1;
}

ITER=3
SF=1
PROG=$KIT_DIR/utils/atranspl
OUT=${OUT_DIR}/atom
USER=${DATABASE_USER}

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

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

echo "Starting Atomicity Test at `date`..."
echo ""
echo "Performing $ITER ACID transactions with COMMIT"
echo ""

$KIT_DIR/utils/randkey $ITER $SF u$USER | $PROG 1 1 1 0 u$USER > ${OUT}c 2>&1

echo "ACID transactions with COMMIT ended. Output in ${OUT}c"
echo ""
echo "Performing $ITER ACID transactions with ROLLBACK"
echo ""

$KIT_DIR/utils/randkey $ITER $SF u$USER | $PROG 1 1 0 0 u$USER > ${OUT}r 2>&1

echo "ACID transactions with ROLLBACK ended. Output in ${OUT}r"
echo ""
echo "Ending Atomicity Test at `date`..."
```

c.4 atranspl.c

```
/* Copyright (c) 2001, 2002, Oracle Corporation.  All rights reserved.  */

/*

   NAME
     atranspl.c  - <one-line expansion of the name>

   DESCRIPTION
      TPC-HR benchmark ACID transaction driver, OCI version 8

   NOTES
      <other useful comments, qualifications, etc.>
```

```
    MODIFIED    (MM/DD/YY)
    mpoess                  10/23/02    -
mpoess_update_from_visa
    mpoess      10/17/01  - add parameter in
ACIDinit
    mpoess      02/22/01  - enlarge timing
array
    mpoess      01/04/01  - Creation

*/

#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

#include "atranspl.h"

/* Declare error handling functions */

double gettime();
void sql_error();
void usage();
void ACIDinit();
void ACIDexit();
int atoi();
void srand48();
long lrand48();

/* declarations for ORDERS */

int o_key = 0;
double o_tprice = 0.0;
double o_newtprice = 0.0;

/* declarations for LINEITEM */

int l_key = 0;
int l_pkey = 0;
int l_skey = 0;

int l_quan = 0;
int l_newquan = 0;
double l_eprice = 0.0;
double l_neweprice = 0.0;
double l_disc = 0.0;
double l_tax = 0.0;

sb2 l_npricei;

/* other declarations */

int delta = 0;
double rprice;
double cost;

int proc_no = 1;        /* process number,
global              */
int num_streams = 1;        /* number of
transaction streams     */
int trig = 0;           /* Trigger Time
*/
int slp = 0;            /* Sleep Time
*/

int logfile;            /* fdes for logfile
for durability (optional)   */
int outfile = 1;            /* output file
(optional)              */
#ifdef LINUX
  FILE *infile;     /* input file (optional)
*/
#else
   FILE *infile = stdin;    /* input file
(optional)              */
                        /* in the
format of <o_key> <delta>   */
#endif
char    lname[UNAME_LEN];           /*
username/passwd combo           */
char *passwd;               /* pointer to
password            */

char buf[WRITE_BUF_LEN];  /* buffer to
write               */

unsigned flag = (unsigned) 0;       /* flag to
```

```
store all sorts of options  */

#define INFILE   0x01u
#define OUTFILE 0x02u
#define LOGFILE 0x04u
#define COMMIT   0x08u
#define DELTA    0x10u

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

int num_iter = 0;                /* number of
iterations                  */

time_t curr_time;               /* Current Time
*/

/* OCI handles */

OCIEnv *tpcenv = NULL;
OCIServer *tpcsrv = NULL;
OCIError *errhp = NULL;
OCISvcCtx *tpcsvc = NULL;
OCISession *tpcusr = NULL;
OCIStmt *curi = NULL;
OCIStmt *curr = NULL;
OCIStmt *cure1 = NULL;
OCIStmt *cure2 = NULL;

/* OCI bind handles */

#ifdef NOLKEY
OCIBind *l_keyi_bp = NULL;
OCIBind *o_keyi_bp = NULL;
#endif /* NOLKEY */

OCIBind *l_key_bp = NULL;
OCIBind *o_key_bp = NULL;
OCIBind *delta_bp = NULL;
OCIBind *l_pkey_bp = NULL;
OCIBind *l_skey_bp = NULL;
OCIBind *l_quan_bp = NULL;
OCIBind *l_newquan_bp = NULL;
OCIBind *l_tax_bp = NULL;
OCIBind *l_disc_bp = NULL;
OCIBind *l_eprice_bp = NULL;
OCIBind *l_neweprice_bp = NULL;
OCIBind *o_tprice_bp = NULL;
OCIBind *o_newtprice_bp = NULL;
OCIBind *rprice_bp = NULL;
OCIBind *cost_bp = NULL;

OCIBind *l_neweprice1_bp = NULL;
OCIBind *l_newquan1_bp = NULL;
OCIBind *o_key1_bp = NULL;
OCIBind *l_key1_bp = NULL;

OCIBind *o_newtprice2_bp = NULL;
OCIBind *o_key2_bp = NULL;

sword status = OCI_SUCCESS;  /* OCI
return value */

char sqlstmt[1024];

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

void usage()
{

   fprintf (stderr,"\nUsage: atrans.o[st]t
<proc_no> <num_streams> <commit>
<delta>\n[i<pathname for input>]
[o<pathname for output>] [d<pathname for
durability file>] [u<uid/passwd>] \n\n");

    fprintf(stderr," proc_no :the
process number within this ACID\n");
   fprintf(stderr," num_streams :the total
number of ACID transaction streams \n");
    fprintf(stderr," commit :1 to
commit transaction, abort otherwise\n\n");
    fprintf(stderr," delta :1 to
generate new random delta, otherwise obtain
delta from input\n\n");
   fprintf(stderr," OPTIONAL
```

```
PARAMETERS:\n");
   fprintf(stderr," i<pathname for input>  :full path name for input file - default is stdin\n");
   fprintf(stderr," o<pathname for output>  :full path name for output file - default is stdout\n");
   fprintf(stderr," d<pathname for durability> :full path name for durability success file - must specify for durability test\n");
   fprintf(stderr," u<uid/passwd>  :Username/Password string - default is tcpd/tpcd\n");
   fprintf(stderr," t<trigger>  :Trigger Time - sleep <trigger> seconds before start\n\n");
   fprintf(stderr," s<sleep>  :Sleep Time - sleep <sleep> seconds before commit or rollback\n\n");
   exit(-1);

}


void ACIDexit() {

   OCILogoff(tpcsvc,errhp);
   OCIhfree(tpcenv,OCI_HTYPE_STMT);
   OCIhfree(tpcsvc,OCI_HTYPE_SVCCTX);
   OCIhfree(tpcsrv,OCI_HTYPE_SERVER);
   OCIhfree(tpcusr,OCI_HTYPE_SESSION);

}



/* type: 0 if environment handle is passed, 1 if error handle is passwd */

void sql_error(errhp,status,type)
      OCIError *errhp;
      sword status;
      sword type;
{
  char msg[2048];
  ub4 errcode;
  ub4 msglen;
  int i,j;

  switch(status) {
  case OCI_SUCCESS_WITH_INFO:
     fprintf(stderr, "Error: Statement returned with info.\n");
     if (type)
        (void) OCIErrorGet(errhp,1,NULL,(sb4*) &errcode, (text*) msg,
                                   2048, OCI_HTYPE_ERROR);
     else
        (void) OCIErrorGet(errhp,1,NULL,(sb4*) &errcode, (text*) msg,
                                   2048, OCI_HTYPE_ENV);
     fprintf(stderr,"%s\n",msg);
     break;
  case OCI_ERROR:
    fprintf(stderr, "Error: OCI call error.\n");
     if (type)
        (void)  OCIErrorGet(errhp,1,NULL, (sb4 *) &errcode, (text*) msg,

2048,OCI_HTYPE_ERROR);
     else
        (void)  OCIErrorGet(errhp,1,NULL, (sb4 *) &errcode, (text*) msg,

2048,OCI_HTYPE_ENV);
     fprintf(stderr,"%s\n",msg);
     break;
  case OCI_INVALID_HANDLE:
    fprintf(stderr, "Error: Invalid Handle.\n");
     if (type)
        (void)  OCIErrorGet(errhp,1,NULL, (sb4 *) &errcode, (text*) msg,
```

```
2048,OCI_HTYPE_ERROR);
      else
         (void)       OCIErrorGet(errhp,1,NULL,
(sb4 *) &errcode, (text*) msg,

2048,OCI_HTYPE_ENV);
      fprintf(stderr,"%s\n",msg);
      break;
   }
  /* Rollback just in case */

   (void)
OCITransRollback(tpcsvc,errhp,OCI_DEFAU
LT);

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

    ACIDexit();

   exit(1);
}

#ifdef LINUX
int main(argc,argv)
#else
void main(argc,argv)
#endif
       int argc;
       char *argv[];
{

   int i;
   char line[64];
   ub4 errcode;
   char msg[2048];
  int need_commit = 0;

  /* Initialize some variables */
#ifdef LINUX
    infile=fopen("/dev/stdin","r");
#endif
   strcpy((char *) lname, "tpcd/tpcd");

 if ((argc > 10) || (argc < 5)) {
     usage();
  }

 /* argv[1] -- Process Number */

 proc_no = atoi(argv[1]);

 /* argv[2] -- Number of Streams */

 num_streams = atoi(argv[2]);

 /* argv[3] -- Commit? */

 if (atoi(argv[3]) == 1)
    BIS(flag, COMMIT);

 /* argv[4] -- Delta? */

 if (atoi(argv[4]) == 1)
    BIS(flag, DELTA);

 /* Process  optional parameters */

  argc -= 4;
  argv += 4;

   while(--argc) {
     ++argv;
    switch(argv[0][0]) {
    case 'u':
        strncpy((char *) lname, ++(argv[0]),
UNAME_LEN);
       if (strchr((char *) lname, '/') == NULL)
{
           fprintf(stderr, "Login name must be
in the format of userid/passwd\n");
           usage();
           exit(-1);
        }
        break;
    case 'i':
        if ((infile = fopen(++(argv[0]), "r")) ==
```

```
NULL) {
          fprintf(stderr,"Cannot open input file %s\n", argv[0]);
          fprintf(stderr,"%s\n",strerror(errno));
          exit(-1);
        }
        BIS(flag, INFILE);
        break;
      case 'o':
        if ((outfile = open(++(argv[0]), (O_RDWR | O_SYNC | O_CREAT), S_IRWXU)) == -1) {
          fprintf(stderr,"Cannot open output file %s\n", argv[0]);
          fprintf(stderr,"%s\n",strerror(errno));
          exit(-1);
        }
        BIS(flag, OUTFILE);
        break;
      case 'd':
        if ((logfile = open(++(argv[0]), (O_RDWR | O_SYNC | O_CREAT), S_IRWXU)) == -1) {
          fprintf(stderr,"Cannot open durability success file %s\n", argv[0]);
          fprintf(stderr,"%s\n",strerror(errno));
          exit(-1);
        }
        BIS(flag, LOGFILE);
        break;
      case 'b':
        num_iter = atoi(++(argv[0]));
        break;
      case 't':
        trig = atoi(++(argv[0]));
        break;
      case 's':
        slp = atoi(++(argv[0]));
        break;
      default:
        fprintf(stderr, "Unknown argument %s\n", argv[0]);
        usage();
        break;
    }
  }


FPRTF(outfile,"------------------------------------------\n");

  /* Initialize the cursors etc. */

  (void) ACIDinit();

  /* sleep for some time (triggering) */

   sleep(trig);

  /* start doing the A CID transactions */

  tr_start = gettime();

   /* The number of iteration we will run depends on the number of */
   /*                input                lines */

  while (fgets(line, 64, infile) != NULL) {

#ifdef NOLKEY
      sscanf(line, "%d %d\n", &o_key, &delta);

    /* Obtain l_key from l_key query */

      OCIsexec(tpcsvc,curi,errhp,1);

       /* l_key is the highest l_linenumber available.  We need to pick */
      /* at random a number between 1..l_key. */

    l_key = (int) ((lrand48() % l_key) + 1);
#else
       sscanf(line, "%d %d %d\n", &o_key, &l_key, &delta);
#endif /* NOLKEY */
```

```
    /* Generate delta if necessary */

    if (BIT(flag, DELTA))
      delta = (int) (floor((drand48() * 100)) +
1);

     /* Now, we are ready to run the ACID
transaction. */

    curr_time = time(NULL);

     FPRTF2(outfile, "Starting ACID
transaction %d at %s...\n", (++num_iter),
                ctime(&curr_time));

     FPRTF1(outfile, "o_key: %d\n", (int)
o_key);
     FPRTF1(outfile, "l_key: %d\n", (int)
l_key);
     FPRTF1(outfile, "delta: %d\n", (int)
delta);

     OCIsexec(tpcsvc,curr,errhp,1);

    curr_time = time(NULL);

    if (!BIT(flag, LOGFILE)) {
     FPRTF1(outfile, "BEFORE
COMMIT/ROLLBACK TRANSACTION at
%s\n", ctime(&curr_time));
     FPRTF1(outfile, "l_extendedprice:
%.2f\n", l_eprice);
     FPRTF1(outfile, "l_quantity:
%d\n", (int) l_quan);
     FPRTF1(outfile, "o_totalprice:
%.2f\n\n", o_tprice);
     }

     FPRTF1(outfile, "Sleep %d seconds
before COMMIT/ROLLBACK...\n\n", slp);
     sleep(slp);

    /* Shall we commit? */
    if (BIT(flag, COMMIT)) {
      need_commit = 1;
      while (need_commit) {

if((status=OCITransCommit(tpcsvc,errhp,OCI
_DEFAULT)) != OCI_SUCCESS) {
                OCIrol(tpcsvc,errhp);

OCIsexec(tpcsvc,curr,errhp,1);
          } else {
              need_commit = 0;
              curr_time = time(NULL);
               FPRTF2(outfile, "ACID
Transaction iteration %d COMMITED at
%s\n",
                   num_iter,
ctime(&curr_time));
           }
       }
     } else {
       OCIrol(tpcsvc,errhp);
       curr_time = time(NULL);
        FPRTF2(outfile, "ACID Transaction
iteration %d ROLLBACK at %s\n",
              num_iter, ctime(&curr_time));
     }

      /* Report all results to outfile and if
necessary, to success file. */

      /* Report initial and new values for
o_totalprice, l_extendedprice, */
     /*                          l_quantity.
*/

     /*
    curr_time = time(NULL);
     FPRTF1(outfile, "Transaction Comp leted
at %s\n", ctime(&curr_time));
     */

      /* Get the values in LINEITEM and
ORDERS after the transaction */
```

```
    if (BIT(flag, LOGFILE)) {
        FPRTF1(logfile, "p_key: %d\n",
(int) l_pkey);
        FPRTF1(logfile, "s_key: %d\n",
(int) l_skey);
        FPRTF1(logfile, "o_key: %d\n",
(int) o_key);
        FPRTF1(logfile, "l_key: %d\n",
(int) l_key);
        FPRTF1(logfile, "delta: %d\n",
(int) delta);
        FPRTF1(logfile, "Transaction
Completed at %s\n", ctime(&curr_time));
        FPRT F(logfile,
"-------------------------------------------------\n");
    } else {

       OCIsexec(tpcsvc,cure1,errhp,1);
       OCIsexec(tpcsvc,cure2,errhp,1);

        FPRTF(outfile, "AFTER
TRANSACTION:\n");
        FPRTF1(outfile, "l_extendedprice:
%.2lf\n", l_neweprice);
        FPRTF1(outfile, "l_quantity:
%d\n", (int) l_newquan);
        FPRTF1(outfile, "o_totalprice:
%.2lf\n\n", o_newtprice);
        FPRTF1(outfile, "l_tax:
%.2lf\n", l_tax);
        FPRTF1(outfile, "l_discount:
%.2lf\n", l_dis c);
        FPRTF1(outfile, "rprice:
%.2lf\n", rprice);
        FPRTF1(outfile, "cost:
%.2lf\n", cost);
       FPRTF(outfile,
"-------------------------------------------------\n")
;
     }
  }

 tr_end = gettime();

  if (!BIT(flag,LOGFILE)) {
     FPRTF1(outfile, "Start Time: %.2f\n",
tr_start);
     FPRTF1(outfile, "End Time: %.2f\n",
tr_end);
    FPRTF1(outfile, "Elapsed Time: %.2f\n",
(tr_end - tr_start));
     FPRTF1(outfile, "Transaction Count:
%d\n", num_iter);
     FPRTF1(outfile, "Transaction Rate:
%.2f\n", num_iter/(tr_end - tr_start));
  } else {
     FPRTF1(logfile, "Start Time: %.2f\n",
tr_start);
     FPRTF1(logfile, "End Time: %.2f\n",
tr_end);
    FPRTF1(logfile, "Elapsed Time: %.2f\n",
(tr_end - tr_start));
     FPRTF1(logfile, "Transaction Count:
%d\n", num_iter);
   }

  /*  Disconnect from ORACLE. */

  if (BIT(flag, INFILE))
     fclose(infile);
  if (BIT(flag, OUTFILE))
     close(outfile);
  if (BIT(flag, LOGFILE))
     close(logfile);

   ACIDexit();

   exit(0);
}


void ACIDinit()
{

  /* run random seed */

   srand48(getpid());
```

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

   (void) OCIInitialize(OCI_DEFAULT,(dvoid
*)0,0,0,0);
   if((status=OCIEnvInit((OCIEnv
**)&tpcenv,OCI_DEFAULT,0,(dvoid
**)0)) !=
        OCI_SUCCESS)
     sql_error(tpcenv, status, 0);


OCIhalloc(tpcenv,&errhp,OCI_HTYPE_ERR
OR);

OCIhalloc(tpcenv,&curi,OCI_HTYPE_STMT)
;

OCIhalloc(tpcenv,&curr,OCI_HTYPE_STMT)
;

OCIhalloc(tpcenv,&cure1,OCI_HTYPE_STM
T);

OCIhalloc(tpcenv,&cure2,OCI_HTYPE_STM
T);

OCIhalloc(tpcenv,&tpcsvc,OCI_HTYPE_SVC
CTX);

OCIhalloc(tpcenv,&tpcsrv,OCI_HTYPE_SER
VER);

OCIhalloc(tpcenv,&tpcusr,OCI_HTYPE_SESS
ION);

  /* Disables auto commit */
/*
  if (ocof(&tpclda)) {
     sql_error(&tpclda, &tpclda);
      ologof(&tpclda);
      exit(-1);
    }
*/

  /* get username and password */

  passwd = strchr(lname, '/');
  *passwd = '\0';
   passwd++;

   if                ((status                =
OCIServerAttach(tpcsrv,errhp,(text
*)0,0,OCI_DEFAULT)) != OCI_SUCCESS)
       sql_error(errhp,status,1);


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcsrv
,0,OCI_ATTR_SERVER,errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,lname
,strlen(lname),OCI_ATTR_USERNAME,
            errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,passw
d,strlen(passwd),OCI_ATTR_PASSWORD,
            errhp);

  if ((status = OCISessionBegin(tpcsvc, errhp,
tpcusr, OCI_CRED_RDBMS,

OCI_DEFAULT)) != OCI_SUCCESS)
       sql_error(errhp,status,1);


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcusr
,0,OCI_ATTR_SESSION,errhp);

  /* Enable session parallel dml */

  sprintf((char *) sqlstmt, PDMLTXT);
   OCIStmtPrepare(curi,errhp,(text
*)sqlstmt,strlen((char *)sqlstmt),

OCI_NTV_SYNTAX,OCI_DEFAULT);
   OCIsexec(tpcsvc,curi,errhp,1);
```

```
  /* Enable session parallel ddl */

  /*sprintf((char *) sqlstmt, PDDLTXT);
   OCIStmtPrepare(curi,errhp,(text
*)sqlstmt,strlen((char *)sqlstmt),

OCI_NTV_SYNTAX,OCI_DEFAULT);
   OCIsexec(tpcsvc,curi,errhp,1);*/


  /* Make session serializable */

   sprintf ((char *) sqlstmt, ISOTXT);
   OCIStmtPrepare(curi,errhp,(text
*)sqlstmt,strlen((char *)sqlstmt),

OCI_NTV_SYNTAX,OCI_DEFAULT);
   OCIsexec(tpcsvc,curi,errhp,1);

  /* Set optimizer_index_cost_adj = 25 */

  sprintf ((char *) sqlstmt, OICATXT);
   OCIStmtPrepare(curi,errhp,(text
*)sqlstmt,strlen((char *)sqlstmt),

OCI_NTV_SYNTAX,OCI_DEFAULT);
   OCIsexec(tpcsvc,curi,errhp,1);

  curr_time = time(NULL);
   printf("\nConnected to ORACLE as user:
%s at %s\n\n", lname, ctime(&curr_time));


#ifdef NOLKEY
    /* Open and Parse cursor for query to
choose determine l_key. */
   /*  Binds  l_key  to  :l_key.
*/

  sprintf((char *) sqlstmt,SQLTXT1);

OCIStmtPrepare(curi,errhp,sqlstmt,strlen((char
*)sqlstmt),OCI_NTV_SYNTAX,OCI_DEFAU
LT);


OCIbbname(curi,&l_keyi_bp,errhp,":l_key",A
DR(l_key),SIZ(l_key),SQLT_INT);

OCIbbname(curi,&o_keyi_bp,errhp,":o_key",
ADR(o_key),SIZ(o_key),SQLT_INT);

#endif /* NOLKEY */

    /* Open and Parse cursor for the ACID
transaction.          */

  sprintf((char *) sqlstmt,SQLTXT2);
   OCIStmtPrepare(curr,errhp,(text
*)sqlstmt,strlen((char *)sqlstmt),

OCI_NTV_SYNTAX,OCI_DEFAULT);

  /* bind variables */


OCIbbname(curr,l_key_bp,errhp,":l_key",AD
R(l_key),SIZ(l_key),SQLT_INT);

OCIbbname(curr,o_key_bp,errhp,":o_key",AD
R(o_key),SIZ(o_key),SQLT_INT);

OCIbbname(curr,delta_bp,errhp,":delta",ADR(
delta),SIZ(delta),SQLT_INT);

OCIbbname(curr,l_pkey_bp,errhp,":l_pkey",A
DR(l_pkey),SIZ(l_pkey),SQLT_INT);

OCIbbname(curr,l_skey_bp,errhp,":l_skey",A
DR(l_skey),SIZ(l_skey),SQLT_INT);

OCIbbname(curr,l_quan_bp,errhp,":l_quan",A
DR(l_quan),SIZ(l_quan),SQLT_INT);

OCIbbname(curr,l_newquan_bp,errhp,":l_new
quan",ADR(l_newquan),
```

```
            SIZ(l_newquan),SQLT_INT);

OCIbbname(curr,l_tax_bp,errhp,":l_tax",ADR(
l_tax),SIZ(l_tax),SQLT_FLT);

OCIbbname(curr,l_disc_bp,errhp,":l_disc",AD
R(l_disc),SIZ(l_disc),SQLT_FLT);

OCIbbname(curr,l_eprice_bp,errhp,":l_eprice",
ADR(l_eprice),SIZ(l_eprice),
            SQLT_FLT);

OCIbbname(curr,l_neweprice_bp,errhp,":l_ne
weprice",ADR(l_neweprice),
            SIZ(l_neweprice),SQLT_FLT);


OCIbbname(curr,o_tprice_bp,errhp,":o_tprice"
,ADR(o_tprice),SIZ(o_tprice),
            SQLT_FLT);

OCIbbname(curr,o_newtprice_bp,errhp,":o_ne
wtprice",ADR(o_newtprice),
            SIZ(o_newtprice),  SQLT_FLT);

OCIbbname(curr,rprice_bp,errhp,":rprice",AD
R(rprice),SIZ(rprice), SQLT_FLT);

OCIbbname(curr,cost_bp,errhp,":cost",ADR(c
ost),SIZ(cost), SQLT_FLT);

  /* Open & Parse cursor for end values query
*/

   sprintf((char *) sqlstmt,SQLTXT3);
   OCIStmtPrepare(cure1,errhp,(text
*)sqlstmt,strlen((char *)sqlstmt),

OCI_NTV_SYNTAX,OCI_DEFAULT);

  sprintf((char *) sqlstmt,SQLTXT4);
   OCIStmtPrepare(cure2,errhp,(text
*)sqlstmt,strlen((char *)sqlstmt),
OCI_NTV_SYNTAX,OCI_DEFAULT);

  /* bind variables */


OCIbbname(cure1,l_neweprice1_bp,errhp,":l_
neweprice",ADR(l_neweprice),
            SIZ(l_neweprice),SQLT_FLT);

OCIbbname(cure1,l_newquan1_bp,errhp,":l_n
ewquan",ADR(l_newquan),
            SIZ(l_newquan),SQLT_INT);

OCIbbname(cure1,o_key1_bp,errhp,":o_key",
ADR(o_key),SIZ(o_key),SQLT_INT);

OCIbbname(cure1,l_key1_bp,errhp,":l_key",A
DR(l_key),SIZ(l_key),SQLT_INT);


OCIbbname(cure2,o_newtprice2_bp,errhp,":o_
newtprice",ADR(o_newtprice),
            SIZ(o_newtprice),SQLT_FLT);

OCIbbname(cure2,o_key2_bp,errhp,":o_key",
ADR(o_key),SIZ(o_key),SQLT_INT);

}
```

c.5 atranspl.h

```
/*  Copyright  (c)  2001,  2002,  Oracle
Corporation.  All rights reserved.  */

/*
   NAME
      atranspl.h  - <one-line expansion of the
name>
```

```
   DESCRIPTION

    MODIFIED    (MM/DD/YY)
     mpoess                 10/23/02 -
mpoess_update_from_visa
      mpoess          10/17/01 - add TXT
parameter
    mpoess       04/09/01 - add hint to find
max linenumber
    mpoess       01/04/01 - Creation

*/
#ifndef ATRANSPL_H

#define ATRANSPL_H

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

#include <oratypes.h>
#ifndef OCIDFN
#include <ocidfn.h>
#endif /* OCIDFN */

#ifndef OCI_ORACLE
#include <oci.h>
#endif /* OCI_ORACLE */

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

extern int errno;

#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 UNAME_LEN 64
#define WRITE_BUF_LEN 1024

#define  NA              -1      /* ANSI
SQL NULL */
#define  VER7            2
#define  NOT_SERIALIZABLE    8177    /*
ORA-08177: transaction not serializable */
#define WRITE_BUF_LEN 1024

#define ADR(object) ((ub1 *)&(object))
#define SIZ(object) ((sword)sizeof(object))
#define  BIS(flg,mask)  (unsigned)  (flg  |=
(unsigned) mask)
#define  BIT(flg,mask)  (unsigned)  ((unsigned)
flg & (unsigned) mask)

#define FPRTF(fd,s) \
{sprintf(buf,s); write(fd, buf, strlen(s));}
#define FPRTF1(fd,s,p) \
{sprintf(buf,s,p); write(fd, buf, strlen(buf));}
#define FPRTF2(fd,s,p1,p2) \
{sprintf(buf,s,p1,p2);      write(fd,      buf,
strlen(buf));}
```

```
#define OCIhalloc(envh,hndl,htyp) \
     if((status=OCIHandleAlloc((dvoid
*)envh,(dvoid **)hndl,htyp,0,(dvoid
**)0))!=OCI_SUCCESS) \
         sql_error(envh,status,0); \
    else \
        DISCARD 0

#define OCIhfree(hndl,htyp) \
     if((status=OCIHandleFree((dvoid
*)hndl,htyp)) == OCI_SUCCESS) \
        fprintf(stderr, "Error freeing handle of
type %d\n", htyp)

#define  OCIaget(hndl,htyp,attp,size,atyp,errh)
\
     if((status=OCIAttrGet((dvoid
*)hndl,htyp,(dvoid *)attp,(dvoid
*)size,atyp,errh)) != OCI_SUCCESS) \
         sql_error(errh,status,1); \
    else \
        DISCARD 0

#define OCIaset(hndl,htyp,attp,size,atyp,errh) \
     if((status=OCIAttrSet((dvoid
*)hndl,htyp,(dvoid *)attp,size,atyp,errh)) !=
OCI_SUCCESS) \
         sql_error(errh,status,1); \
    else \
        DISCARD 0

#define OCIsexec(svch,stmh,errh,iter) \

if((status=OCIStmtExecute(svch,stmh,errh,iter,
0,NULL,NULL,OCI_DEFAULT)) !=
OCI_SUCCESS) \
         sql_error(errh,status,1); \
    else \
        DISCARD 0

#define
OCIbbname(stmh,bindp,errh,sqlvar,progv,prog
vl,ftype) \

if((status=OCIBindByName(stmh,&bindp,errh
,(text *)sqlvar,strlen(sqlvar), \

progv,progvl,ftype,0,0,0,0,0,OCI_DEFAULT))
!= OCI_SUCCESS) \
         sql_error(errh,status,1); \
    else \
        DISCARD 0

#define
OCIbbnamei(stmh,bindp,errh,sqlvar,progv,pro
gvl,ftype,indp) \
     if((status=OCIHandleAlloc((dvoid
*)stmh,(dvoid **)&bindp,OCI_HTYPE_BIND,
\
                                    0,(dvoid
**)0))!=OCI_SUCCESS) \
         sql_error(stmh,status,0); \

if((status=OCIBindByName(stmh,&bindp,errh
,(text *)sqlvar,strlen(sqlvar), \

progv,progvl,ftype,indp,0,0,0,0,OCI_DEFAUL
T)) != OCI_SUCCESS) \
         sql_error(errh,status,1); \
    else \
        DISCARD 0

#define OCIcom(svcp,errh) \

if((status=OCITransCommit(svcp,errh,OCI_D
EFAULT)) != OCI_SUCCESS) \
          sql_error(errh,status,1); \
      else \
          DISCARD 0

#define OCIrol(svcp,errh) \

if((status=OCITransRollback(svcp,errh,OCI_D
EFAULT)) != OCI_SUCCESS) \
          sql_error(errh,status,1); \
      else \
          DISCARD 0
```

```
#define ISOTXT "alter session set isolation_level = serializable"
#define PDMLTXT "alter session force parallel dml parallel (degree 2)"
#define PDDLTXT "alter session force parallel ddl parallel (degree 2)"
#define OICATXT "alter session set optimizer_index_cost_adj=25"

#define SQLTXT1 "BEGIN SELECT /*+ index(lineitem,i_l_orderkey) */ MAX(l_linenumber) INTO :l_key FROM lineitem \
WHERE l_orderkey = :o_key; END;"

#define SQLTXT2 "BEGIN d_atrans.doatrans(:l_key, :o_key, :delta, :l_pkey, \
:l_skey, :l_quan, :l_newquan, :l_tax, :l_disc, :l_eprice, :l_neweprice, \
:o_tprice, :o_newtprice, :rprice, :cost); END;"

#define SQLTXT3 "BEGIN SELECT l_extendedprice, l_quantity \
INTO :l_neweprice, :l_newquan \
FROM lineitem \
WHERE l_orderkey = :o_key \
AND l_linenumber = :l_key; END;"

#define SQLTXT4 "BEGIN SELECT o_totalprice INTO :o_newtprice \
FROM orders \
WHERE o_orderkey = :o_key; END;"

#define SQLTXT5 "BEGIN SELECT l_extendedprice, l_quantity \
INTO :l_eprice, :l_quan \
FROM lineitem \
WHERE l_orderkey = :o_key \
AND l_linenumber = :l_key; END;"

#define SQLTXT6 "BEGIN SELECT o_totalprice INTO :o_tprice \
FROM orders \
WHERE o_orderkey = :o_key; END;"

#endif /* ATRANSPL_H */
```

c.6 atrans.sql

```
Rem
Rem $Header: atrans.sql 07-aug-99.21:27:13 mpoess Exp $
Rem
Rem atrans.sql
Rem
Rem  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
Rem
Rem      NAME
Rem        atrans.sql  - <one-line expansion of the name>
Rem
Rem     DESCRIPTION
Rem                Creates ACID Transaction Package for TPC-D benchmark.
Rem                Asks user to input values for o_key, delta and output file.
Rem
Rem     NOTES
Rem                <other useful comments, qualifications, etc.>
Rem
Rem      MODIFIED     (MM/DD/YY)
Rem      mpoess       08/07/99  - Creation
Rem      mpoess       08/07/99  - Created
Rem


set serverout on;
set termout on;
set echo on;

CREATE OR REPLACE PACKAGE d_atrans
```

```
IS
PROCEDURE doatrans
(
      l_key             IN OUT integer,
      o_key             IN OUT integer,
      delta       IN OUT integer,
      l_pkey             IN OUT integer,
      l_skey             IN OUT integer,
      l_quan             IN OUT integer,
      l_newquan          IN OUT integer,
      l_tax              IN OUT number,
      l_disc             IN OUT number,
      l_eprice           IN OUT number,
      l_neweprice        IN OUT number,
      o_tprice    IN OUT number,
      o_newtprice     IN OUT number,
      rprice             IN OUT number,
      cost        IN OUT number
);
END;
/

CREATE OR REPLACE PACKAGE BODY
d_atrans
IS
PROCEDURE doatrans
(
      l_key             IN OUT integer,
      o_key             IN OUT integer,
      delta             IN OUT integer,
      l_pkey             IN OUT integer,
      l_skey             IN OUT integer,
      l_quan             IN OUT integer,
      l_newquan          IN OUT integer,
      l_tax              IN OUT number,
      l_disc             IN OUT number,
      l_eprice           IN OUT number,
      l_neweprice        IN OUT number,
      o_tprice    IN OUT number,
      o_newtprice     IN OUT number,
      rprice             IN OUT number,
      cost        IN OUT number
)
IS
            ototal number;
            not_serializable EXCEPTION;
             PRAGMA
EXCEPTION_INIT(not_serializable,-8177);
BEGIN
  LOOP BEGIN

      select o_totalprice
       into o_tprice
       from orders
       where o_orderkey = o_key;

      select      l_quantity,      l_extendedprice,
l_partkey, l_suppkey, l_tax, l_discount
            into l_quan, l_eprice, l_pkey, l_skey,
l_tax, l_disc
            from lineitem
            where l_orderkey = o_key
            and     l_linenumber = l_key;

     ototal := o_tprice  - trunc((trunc((l_eprice
* (1.0-l_disc)),2) * (1.0+l_tax)),2);
     rprice := trunc((l_eprice/l_quan), 2);
    cost := trunc((rprice * delta), 2);
    l_neweprice := l_eprice + cost;
     o_newtprice := trunc((l_neweprice * (1.0
- l_disc)), 2);
      o_newtprice          :=          ototal          +
trunc((o_newtprice * (1.0 + l_tax)), 2);
    l_newquan := l_quan + delta;

     update lineitem
         set l_extendedprice = l_neweprice,
         l_quantity = l_newquan
         where l_orderkey = o_key
         and l_linenumber = l_key;

     update orders
         set o_totalprice = o_newtprice
         where o_orderkey = o_key;

        insert into history (h_p_key, h_s_key,
h_o_key, h_l_key, h_delta, h_date_t)
           values (l_pkey, l_skey, o_key, l_key,
```

```
delta, sysdate);

      EXIT;

   EXCEPTION
     WHEN not_serializable THEN
        ROLLBACK;
   END;

  END LOOP;

END doatrans;
END;
/

exit;
```

c.7 ckpt.sh

```
#!/bin/ksh
#
# $Header: ckpt.sh 08-aug-99.17:32:22 mpoess
Exp $
#
# ckpt.sh
#
#  Copyright (c) Oracle Corporation 1999. All
Rights Reserved.
#
#      NAME
#         ckpt.sh - <one-line expansion of the
name>
#
#      DESCRIPTION
#         <short description of component this
file declares/defines>
#
#      NOTES
#                  <other useful comments,
qualifications, etc.>
#
#      MODIFIED     (MM/DD/YY)
#      mpoess         08/08/99 - Creation
#      mpoess         08/08/99 - Creation
#
. $KIT_DIR/env
sqlplus -s /NOLOG << !

          connect / as sysdba;
          alter system switch logfile;
           alter system switch logfile;
           exit;
!
```

c.8 cnt_hist.sql

```
select count(*) from history;
exit;
```

c.9 consist.sh

```
#!/bin/ksh
#
#  $Header: consist.sh 08-aug-99.14:20:51
mpoess Exp $
#
# consist.sh
#
#  Copyright (c) Oracle Corporation 1999. All
Rights Reserved.
#
#      NAME
#         consist.sh - <one-line expansion of the
```

```
name>
#
#   DESCRIPTION
#        Performs consistency tests.
#         Usage: consist.sh [-n iter] [-s number of stream] [-p prog]
#                                        [-u usr/pswd] -h
#
#        Options:  See usage below
#
#      NOTES
#                    <other useful comments, qualifications, etc.>
#
#      MODIFIED     (MM/DD/YY)
#      mpoess       08/08/99 - Creation
#      mpoess       08/08/99 - Creation
#

. $KIT_DIR/env

OH=$ORACLE_HOME
# ACID_DIR=$TPCD_KIT_DIR/audit set in env
OUT_DIR=$ACID_OUT

KEY=$OUT_DIR/key$$_
OUTFILE=${OUT_DIR}/consrte
CON1=${OUT_DIR}/conb
CON2=${OUT_DIR}/cona
CHK=${OUT_DIR}/consckpt

/bin/rm -rf ${KEY}* $CON1 $CON2 $OUTFILE $CHK

trap "/bin/rm -rf ${KEY}*; exit 1" 1 2 3 15

STREAM=${NUM_STREAMS}
let STREAM="$STREAM + 1" # add one for the update stream
ITER=100
PROG=atranspl
USER=${DATABASE_USER}
CK=10

usage() {

    echo ""
     echo "Usage: $0 [-n iter] [-s number of stream] [-p prog] [-u usr/pswd] -h"
    echo ""
     echo "-n iter                : number of iterations, default is 100"
     echo "-s number of stream : number of streams, default is 2"
     echo "-p prog                : program to run, default is atranspl.ott"
     echo "-u usr/pswd            : user/password for database access, default is tpcd/tpcd"
     echo "-t chkpt               : time after the start of ACID transaction to perform the checkpoint"
     echo "                         default is 10 seconds"
     echo "-h                     : print this usage summary"
    exit 1;
}

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

while :
do
    case "$1" in
     -s) shift; STREAM=$1;;
     -n) shift; ITER=$1;;
     -p) shift; PROG=$1;;
     -u) shift; USER=$1;;
     -t) shift; CK=$1;;
     -h) usage; exit 0;;
     --) break;;
     esac
     shift
done
```

```
if [ $ITER -lt 100 ]
then
echo "Error: Must at least run 100 iterations!"
echo "Exiting..."
exit 1
fi

if [ $STREAM -lt 2 ]
then
echo "Error: Must at least run 2 streams!"
echo "Exiting..."
exit 1
fi

echo "Starting Consistency Test at `date`..."
echo ""
echo "Generate some keys first"
echo ""

i=0

while [ $i -lt $STREAM ]
do
   echo randkey $ITER 1 u$USER
   randkey $ITER 1 u$USER > ${KEY}$i
   i=`expr $i + 1`
done

echo "Check consistency before Submitting Transactions `date`"
echo "Check consistency before Submitting Transactions `date`" >> $CON1

echo "Obtain 10 keys from the each key file to check consistency"

i=0
while [ $i -lt $STREAM ]
do
KEYS=`head -10 ${KEY}${i} | awk '{printf "%d ", $1}'`
echo "The 10 Keys for file $i are: $KEYS"
#for j in `head -10 ${KEY}${i} | awk '{printf "%d ", $1}'`
for j in $KEYS
do
    sqlplus $USER @consist $j >> $CON1
      echo "------------------------------" >> $CON1
done
    i=`expr $i + 1`
done

echo ""
echo "Starting ACID transactions at `date`"
echo ""

i=0

while [ $i -lt $STREAM ]
do
      $PROG $i $STREAM 1 0 u${USER} i${KEY}${i} o${OUTFILE}${i} s1 &
    i=`expr $i + 1`
done

echo "Schedule a Checkpoint"
echo "Checkpoint scheduled at $CK seconds after `date`"

(sleep $CK; $ACID_DIR/ckpt.sh) &

wait

echo ""
echo "Ending ACID transactions at `date`"
echo ""

echo "Completed $STREAM transaction streams with $ITER iterations each"
echo ""

echo "Check consistency after Submitting Transactions `date`"
echo "Check consistency after Submitting Transactions `date`" >> $CON2

cat
```

```
${ORACLE_HOME}/rdbms/log/alert_${ORA
CLE_SID}.log >> $CHK

i=0
while [ $i -lt $STREAM ]
do
KEYS=`head -10 ${KEY}$i | awk '{printf
"%d ", $1}'`
#for j in `head -10 ${KEY}$i | awk '{printf
"%d ", $1}'`
echo "The keys to check for consistency after
the test from file $i are:"
echo "$KEYS"
for j in $KEYS
do
    sqlplus $USER @consist $j >> $CON2
      echo      "------------------------------"      >>
$CON2
done
    i=`expr $i + 1`
done
```

c.10 consist.sql

```
set verify off
rem set termout on
rem set echo on


REM
REM Get today's date.
REM

select
substr(TO_CHAR(sysdate,'YYYY-MM-DD
HH:MI:SS'),1,20) as CURRENT_TIME
from dual;

set serverout on;

DECLARE
            o_okey              number;
            o_tprice            number;
            l_tprice            number;
            diff                number;
BEGIN
     select o_totalprice
          into o_tprice
          from orders
         where o_orderkey = &&1;



     select /*+ index(lineitem,i_l_orderkey) */
sum(trunc((trunc((l_extendedprice                *
(1-l_discount)), 2)
               * (1+l_tax)), 2))
          into l_tprice
          from lineitem
         where l_orderkey = &&1;

     diff := l_tprice - o_tprice;

      dbms_output.put_line('O_TOTALPRICE:
' || TO_CHAR(trunc(o_tprice,2)));
      dbms_output.put_line('L_TOTALPRICE:
' || TO_CHAR(trunc(l_tprice,2)));
       dbms_output.put_line('Difference:     ' ||
TO_CHAR(trunc(diff,2)));

END;
.
/


spool off
exit
```

c.11 count_tx.sh

```
#!/bin/ksh

STEM=$1
ITER=$2
OUT=$3
FIN=FALSE
while [ "$FIN" = "FALSE" ]
do
  s=0
  FIN=TRUE
 while [ $s -lt $STEM ]
  do
     nt=`grep "Transaction Completed" $OUT/dura${s} | wc -l`
    if [ $nt -lt $ITER ];then
        FIN=FALSE
     fi
    s=`expr $s + 1`
  done
  sleep 5
done
echo all streams have commited $ITER transactions
```

c.12 d_hist.sql

```
Rem
Rem $Header: d_hist.sql 07-aug-99.21:33:08 mpoess Exp $
Rem
Rem d_hist.sql
Rem
Rem Copyright (c) Oracle Corporation 1999. All Rights Reserved.
Rem
Rem NAME
Rem d_hist.sql - <one-line expansion of the name>
Rem
Rem DESCRIPTION
Rem Creates a history table for ACID test purpose.
Rem
Rem NOTES
Rem <other useful comments, qualifications, etc.>
Rem
Rem MODIFIED (MM/DD/YY)
Rem mpoess 08/07/99 - Creation
Rem mpoess 08/07/99 - Created
Rem


set termout on;
set serverout on;
set echo on;

drop table history;

create table history
(
    h_p_key number,
    h_s_key number,
    h_o_key number,
    h_l_key number,
    h_delta number,
    h_date_t date
);

exit;
```

c.13 dura.sh

```
#!/bin/ksh
#
# $Header: dura.sh 08-aug-99.15:21:38 mpoess Exp $
#
# dura.sh
#
```

```
# Copyright (c) Oracle Corporation 1999. All
Rights Reserved.
#
#       NAME
#         dura.sh - <one-line expansion of the
name>
#
#       DESCRIPTION
#         <short description of component this
file declares/defines>
#
#       NOTES
#                   <other useful comments,
qualifications, etc.>
#
#       MODIFIED     (MM/DD/YY)
#       mpoess        08/08/99 - Creation
#       mpoess        08/08/99 - Creation
#

. $KIT_DIR/env

# Create history table

# Count number of entries in the history table

SERVER="ultraperf2"

echo
"----------------------------------------------------"
echo "Capturing Process information before
durability tests `date`"
rsh $SERVER -n -l spyda  ps -ef; date
echo
"----------------------------------------------------"

echo
"----------------------------------------------------"
echo "Starting the durability tests `date`"
run_acid.sh &
echo
"----------------------------------------------------"

sleep 1200

echo
"----------------------------------------------------"
echo "Collecting user information. `date`"
./cnt_user.sh pswong spyda ultraperf2 >
dura/duraucnt 2>&1
echo
"----------------------------------------------------"

echo
"----------------------------------------------------"
echo "Capturing Process information while
running Transactions `date`"
rsh $SERVER -n -l spyda  ps -ef; date
echo
"----------------------------------------------------"

echo
"----------------------------------------------------"
echo "Capturing disk information on Server:
Ultraperf2 `date`"
rsh $SERVER -n -l spyda  vxprint -ht ; date
echo
"----------------------------------------------------"

echo
"----------------------------------------------------"
echo "Detaching mirror on data disk. `date`"
rsh $SERVER -n -l root "vxplex -v ordr23
det ordr23-01"
echo
"----------------------------------------------------"

echo
"----------------------------------------------------"
echo "Capturing Disk information information
on Server: Ultraperf2 `date`"
rsh $SERVER -n -l spyda  vxprint -ht ; date
echo
"----------------------------------------------------"

sleep 120

echo
```

```
"----------------------------------------------------"
echo "Capturing Process information after breaking data mirror. `date`"
rsh $SERVER -n -l spyda  ps -ef; date
echo
"----------------------------------------------------"



echo
"----------------------------------------------------"
echo "Detaching mirror on log2 disk. `date`"
rsh $SERVER -n -l root "vxplex -v log2 det log2-01"
echo
"----------------------------------------------------"

echo
"----------------------------------------------------"
echo "Capturing Disk information information on Server: Ultraperf2 `date`"
rsh $SERVER -n -l spyda  vxprint -ht ; date
echo
"----------------------------------------------------"

sleep 120

echo
"----------------------------------------------------"
echo "Capturing Process information after detaching log mirror. `date`"
rsh $SERVER -n -l spyda  ps -ef; date
echo
"----------------------------------------------------"

# Power Off
```

c.14 end_acid.sh

```
#!/bin/ksh
#
# $Header: end_acid.sh 08-aug-99.17:06:20 mpoess Exp $
#
# end_acid.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      end_acid.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#       end_cons.sh <pid of the durability run>
#      Options: See usage below
#
#    NOTES
#                <other useful comments, qualifications, etc.>
#
#    MODIFIED    (MM/DD/YY)
#    mpoess      08/08/99 - Creation
#    mpoess      08/08/99 - Creation
#

. $KIT_DIR/env

OH=$ORACLE_HOME
# ACID_DIR=$OH/tpcd/audit set in env
OUT_DIR=$ACID_OUT/
DURA_DIR=$ACID_OUT/dura
RUN_ID_FILE=$ACID_DIR/run_id

SHELL_PID=`cat ${DURA_DIR}/shellpid`
ITER=100
STEM=${NUM_STREAMS}
let STEM="$STEM + 1" # add one for the update stream
PROG=${ACID_DIR}/atranspl.ott
IN=${ACID_DIR}/acid_in
DURA=${DURA_DIR}/drate
```

```
OUT=${DURA_DIR}/drate
DSMPL=${DURA_DIR}/durasmpl
KEY=${DURA_DIR}/key${SHELL_PID}_
USER=tpch/tpch
TRIG=1
HCNT=duracnta

# get history count

sqlplus $USER @cnt_hist > $DURA_DIR/$HCNT 2>&1

# perform the consistency

i=0
while [ $i -lt $STEM ]
do
    for j in `head -10 ${KEY}${i} | awk '{printf "%d ",$1}'`
    do
        sqlplus tpch/tpch @consist $j >> $DURA_DIR/duraconsa
    done
   i=`expr $i + 1`
done

i=0
while [ $i -lt $STEM ]
do
sample.sh $DURA${i} > ${DSMPL}${i} 2>&1
i=`expr $i + 1`
done
```

c.15 gettime.c

```
#ifdef RCSID
static char *RCSid =
    "$Header: gettime.c 15-jul-99.14:27:44 mpoess Exp $ ";
#endif /* RCSID */

/* Copyright (c) Oracle Corporation 1999. All Rights Reserved. */

/*

    NAME
      gettime.c

    DESCRIPTION
      get wall clock time.
      get cpu time.

    FUNCTIONS
      get wall clock time.
      get cpu time.

    NOTES
     Both routines return time in seconds as a double.
    MODIFIED    (MM/DD/YY)
    mpoess      07/15/99 - Creation
    mpoess      07/15/99 - Creation

*/


/*
**  Options:
**      TIME_W_TIMES:       implement gettime() with times().
**      TIME_W_GETTIME:     implement gettime() with gettimeofday().
**      CPU_W_TIMES:        implement getcpu()  with times().
**      CPU_W_GETRU:        implement getcpu()  with getrusage().
**        GETRU_STATS:          collect getrusage statistics
**        GET_P_STATS:          collect get_process_stats statistics
*/
```

```
#define SUN_OS5

#if defined(SUN_OS5)
#define TIME_W_GETTIME
#define CPU_W_TIMES
#undef GETRU_STATS
#undef CPU_W_GETRU
#endif /* SUN_OS5 */

#if defined(sequent) || defined(SEQ_PSX)
# define GET_P_STATS
#endif /* sequent */

#if defined(aix) || defined(AIXRIOS)
# define TIME_W_GETTIME
# define CPU_W_TIMES
# define GETRU_STATS
#endif /* AIXRIOS */

#if defined(a_osf) || defined(A_OSF)
# define TIME_W_GETTIME
# define CPU_W_GETRU
# define GETRU_STATS
#endif /* AIXRIOS */

#if defined(HPUX) || defined(XENIX_386) ||
defined(SYSV_386) || defined(ATT_3B)
# define TIME_W_TIMES
# define CPU_W_TIMES
#endif /* HPUX || XENIX_386 || SYSV_386
*/

#if !defined(TIME_W_GETTIME)
&& !defined(TIME_W_TIMES)
# define TIME_W_TIMES
#endif

#if !defined(CPU_W_GETRU)
&& !defined(CPU_W_TIMES)
# define CPU_W_TIMES
#endif

#ifdef GET_P_STATS
# ifdef GETRU_STATS
#  undef GETRU_STATS
# endif
#endif

#if defined(TIME_W_GETTIME) ||
defined(CPU_W_GETRU) ||
defined(GETRU_STATS)
# include <sys/time.h>
#endif /* TIME_W_GETTIME ||
CPU_W_GETRU || GETRU_STATS */

#if defined(CPU_W_GETRU) ||
defined(GETRU_STATS)
# include <sys/resource.h>
#endif /* CPU_W_GETRU || GETRU_STATS
*/

#if defined(TIME_W_TIMES) || defined
(CPU_W_TIMES)
# include <sys/types.h>
# include <sys/times.h>
# include <sys/param.h> /* most systems
define HZ here */
#endif /* TIME_W_TIMES or
CPU_W_TIMES */

#ifdef GET_P_STATS
# include <sys/types.h>
# include <sys/procstats.h>
#endif /* GET_P_STATS */

# include <stdio.h>

#ifdef GETRU_STATS
struct rusage selfru;
struct rusage kidsru;
#endif /* GETRU_STATS */

#ifdef GET_P_STATS
struct process_stats selfru;
struct process_stats kidsru;
#endif /* GET_P_STATS */
```

```
double gettime ()

{

#ifdef TIME_W_GETTIME
    struct timeval tv;

   (void) gettimeofday (&tv, (struct timezone
*) 0);
    return ((double) tv.tv_sec + (1.0e-6 *
(double) tv.tv_usec));
#endif /* TIME_W_GETTIME */

#ifdef TIME_W_TIMES
   struct tms buf;

  return ((double) times (&buf) / HZ);
#endif /* TIME_W_TIMES */

}



double getcpu ()

{

#ifdef CPU_W_TIMES
   struct tms buf;

   (void) times (&buf);
   return (((double) buf.tms_utime + (double)
buf.tms_stime) / HZ);
#endif /* CPU_W_TIMES */

#ifdef CPU_W_GETRU
   struct rusage ru;
   double usecs;

  (void) getrusage (0, &ru);
    usecs = 1.0e-6 * (double)
(ru.ru_utime.tv_usec + ru.ru_stime.tv_usec);
    return ((double) (ru.ru_utime.tv_sec +
ru.ru_stime.tv_sec) + usecs);
#endif /* CPU_W_GETRU */

}



getru (fp, kids, config, runname, proc_no)

FILE *fp;
int kids;
char *config;
char *runname;
int proc_no;

{

#ifdef GETRU_STATS
   struct rusage ru;

     fprintf (fp, "%-10.10s %-10.10s %10d
%10d ", config,runname, proc_no, kids);
   getrusage (kids ? RUSAGE_CHILDREN :
RUSAGE_SELF, &ru);
    print_ru (fp, &ru);
   fprintf (fp, "\n");
#endif /* GETRU_STATS */

#ifdef GET_P_STATS
    timeval_t tv;
   struct process_stats ru;

     fprintf (fp, "%-10.10s %-10.10s %10d
%10d ", config,runname, proc_no, kids);
    if (kids)
        get_process_stats (&tv, PS_SELF,
(struct process_stats *) 0, &ru);
     else
       get_process_stats (&tv, PS_SELF, &ru,
(struct process_stats *) 0);
   print_ru (fp, &ru);
   fprintf (fp, "\n");
#endif /* GET_P_STATS */
```

```
}



getru1 (kids)

int kids;

{

#ifdef GETRU_STATS
   if (kids) {
      memset (&kidsru, 0, sizeof (kidsru));
        getrusage (RUSAGE_CHILDREN,
&kidsru);
    }
   else {
      memset (&selfru, 0, sizeof (selfru));
       getrusage (RUSAGE_SELF, &selfru);
    }
#endif /* GETRU_STATS */

#ifdef GET_P_STATS
   timeval_t tv;

   if (kids) {
      memset (&kidsru, 0, sizeof (kidsru));
        get_process_stats (&tv, PS_SELF,
(struct process_stats *) 0, &kidsru);
    }
   else {
      memset (&selfru, 0, sizeof (selfru));
        get_process_stats (&tv, PS_SELF,
&selfru, (struct process_stats *) 0);
    }
#endif /* GET_P_STATS */

}



getru2 (fp, kids, config, runname, proc_no)

FILE *fp;
int kids;
char *config;
char *runname;
int proc_no;

{

#ifdef GETRU_STATS
   struct rusage ru;

     fprintf (fp, "%-10.10s %-10.10s %10d
%10d ", config, runname, proc_no, kids);
   getrusage (kids ? RUSAGE_CHILDREN :
RUSAGE_SELF, &ru);
   if (kids)
      diffru (&ru, &kidsru);
    else
      diffru (&ru, &selfru);
   print_ru (fp, &ru);
   fprintf (fp, "\n");
#endif /* GETRU_STATS */

#ifdef GET_P_STATS
   timeval_t tv;
   struct process_stats ru;

     fprintf (fp, "%-10.10s %-10.10s %10d
%10d ", config, runname, proc_no, kids);
   if (kids)
       get_process_stats (&tv, PS_SELF,
(struct process_stats *) 0, &ru);
    else
       get_process_stats (&tv, PS_SELF, &ru,
(struct process_stats *) 0);
   if (kids)
      diffru (&ru, &kidsru);
    else
      diffru (&ru, &selfru);
   print_ru (fp, &ru);
   fprintf (fp, "\n");
#endif /* GET_P_STATS */

}
```

```
#ifdef GETRU_STATS

print_ru (fp, ru)

FILE *fp;
struct rusage *ru;

{

    fprintf (fp, "%10ld ", ru->ru_utime.tv_sec
* 1000 +

(ru->ru_utime.tv_usec/1000));
   fprintf (fp, "%10ld ", ru->ru_stime.tv_sec *
1000 +

(ru->ru_stime.tv_usec/1000));
   fprintf (fp, "%10ld ", ru->ru_maxrss);
   fprintf (fp, "%10ld ", ru->ru_majflt);
   fprintf (fp, "%10ld ", ru->ru_minflt);
   fprintf (fp, "%10ld ", 0);
   fprintf (fp, "%10ld ", 0);
   fprintf (fp, "%10ld ", 0);
   fprintf (fp, "%10ld ", ru->ru_nswap);
    fprintf (fp, "%10ld ", 0);
   fprintf (fp, "%10ld ", ru->ru_nvcsw);
   fprintf (fp, "%10ld ", ru->ru_nivcsw);
   fprintf (fp, "%10ld ", ru->ru_nsignals);
   fprintf (fp, "%10ld ", 0);
   fprintf (fp, "%10ld ", 0);
   fprintf (fp, "%10ld ", ru->ru_inblock);
   fprintf (fp, "%10ld ", ru->ru_oublock);
   fprintf (fp, "%10ld ", 0);
    fprintf (fp, "%10ld",   0);

}


diffru (ru2, ru)

struct rusage *ru2;
struct rusage *ru;

{

    ru2->ru_utime.tv_sec                -=
ru->ru_utime.tv_sec;
    ru2->ru_utime.tv_usec               -=
ru->ru_utime.tv_usec;
    ru2->ru_stime.tv_sec                -=
ru->ru_stime.tv_sec;
    ru2->ru_stime.tv_usec               -=
ru->ru_stime.tv_usec;
    ru2->ru_maxrss -= ru->ru_maxrss;
    ru2->ru_ixrss -= ru->ru_ixrss;
    ru2->ru_idrss -= ru->ru_idrss;
    ru2->ru_minflt -= ru->ru_minflt;
    ru2->ru_majflt -= ru->ru_majflt;
    ru2->ru_nswap -= ru->ru_nswap;
    ru2->ru_inblock -= ru->ru_inblock;
    ru2->ru_oublock -= ru->ru_oublock;
    ru2->ru_msgsnd -= ru->ru_msgsnd;
    ru2->ru_msgrcv -= ru->ru_msgrcv;
    ru2->ru_nsignals -= ru->ru_nsignals;
    ru2->ru_nvcsw -= ru->ru_nvcsw;
    ru2->ru_nivcsw -= ru->ru_nivcsw;

}

#endif /* GETRU_STATS */



#ifdef GET_P_STATS

print_ru (fp, ps)

FILE *fp;
struct process_stats *ps;

{

    fprintf (fp, "%lu ", ps->ps_utime.tv_sec *
1000 +
```

```
(ps->ps_utime.tv_usec/1000));
    fprintf (fp, "%lu ", ps->ps_stime.tv_sec *
1000 +

(ps->ps_stime.tv_usec/1000));
   fprintf (fp, "%lu ", ps->ps_maxrss);
   fprintf (fp, "%lu ", ps->ps_pagein);
   fprintf (fp, "%lu ", ps->ps_reclaim);
   fprintf (fp, "%lu ", ps->ps_zerofill);
   fprintf (fp, "%lu ", ps->ps_pffincr);
   fprintf (fp, "%lu ", ps->ps_pffdecr);
   fprintf (fp, "%lu ", ps->ps_swap);
   fprintf (fp, "%lu ", ps->ps_syscall);
   fprintf (fp, "%lu ", ps->ps_volcsw);
   fprintf (fp, "%lu ", ps->ps_involcsw);
   fprintf (fp, "%lu ", ps->ps_signal);
   fprintf (fp, "%lu ", ps->ps_lread);
   fprintf (fp, "%lu ", ps->ps_lwrite);
   fprintf (fp, "%lu ", ps->ps_bread);
   fprintf (fp, "%lu ", ps->ps_bwrite);
    fprintf (fp, "%lu ", ps->ps_phread);
    fprintf (fp, "%lu",   ps->ps_phwrite);

}



diffru (ru2, ru)

struct process_stats *ru2;
struct process_stats *ru;

{

    ru2->ps_utime.tv_sec                -=
ru->ps_utime.tv_sec;
    ru2->ps_utime.tv_usec               -=
ru->ps_utime.tv_usec;
    ru2->ps_stime.tv_sec                -=
ru->ps_stime.tv_sec;
    ru2->ps_stime.tv_usec               -=
ru->ps_stime.tv_usec;
    ru2->ps_maxrss -= ru->ps_maxrss;
    ru2->ps_pagein -= ru->ps_pagein;
    ru2->ps_reclaim -= ru->ps_reclaim;
    ru2->ps_zerofill -= ru->ps_zerofill;
    ru2->ps_pffincr -= ru->ps_pffincr;
    ru2->ps_pffdecr -= ru->ps_pffdecr;
    ru2->ps_swap -= ru->ps_swap;
    ru2->ps_syscall -= ru->ps_syscall;
    ru2->ps_volcsw -= ru->ps_volcsw;
    ru2->ps_involcsw -= ru->ps_involcsw;
    ru2->ps_signal -= ru->ps_signal;
    ru2->ps_lread -= ru->ps_lread;
    ru2->ps_lwrite -= ru->ps_lwrite;
    ru2->ps_bread -= ru->ps_bread;
    ru2->ps_bwrite -= ru->ps_bwrite;
    ru2->ps_phread -= ru->ps_phread;
    ru2->ps_phwrite -= ru->ps_phwrite;

}

#endif /* GET_P_STATS */
```

c.16 gtime.c

```
/* Copyright (c) 2001, 2002, Oracle
Corporation.  All rights reserved.  */

/*

    NAME
      gtime.c - <one-line expansion of the
name>

    DESCRIPTION
       <short description of facility this file
declares/defines>

   EXPORT FUNCTION(S)
       <external functions defined for use
outside package - one-line descriptions>

   INTERNAL FUNCTION(S)
```

```
        <other external functions defined - one-line descriptions>

    STATIC FUNCTION(S)
        <static functions defined - one-line descriptions>

    NOTES
        <other useful comments, qualifications, etc.>

    MODIFIED    (MM/DD/YY)
     mpoess                 10/23/02 - mpoess_update_from_visa
    mpoess          08/29/01 - Creation

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

# include <sys/time.h>

main ()
{

  struct timeval tv;

            (void) gettimeofday (&tv, (struct timezone *) 0);

   printf ("%.2f\n", ((double) tv.tv_sec + (1.0e-6 * (double) tv.tv_usec)) ) ;

}

/* end of file gtime.c */
```

c.17 iso1.sh

```
#!/bin/ksh
#
# $Header: iso1.sh 29-jul-98.17:00:11 akarasik Exp $
#
# iso1.sh
#
#   Copyright (c) Oracle Corporation 1998. All Rights Reserved.
#
#       NAME
#         iso1.sh
#
#   DESCRIPTION
#           Usage: iso1.sh  [-u user/password] [-n remote_node] -h
#          Options: See usage below
#   NOTES
#      For a cross node isolation test, assume the local node is
#      one of the participating nodes.  The other node can be
#      specified by the -n option.
#            You need to set the environment variable TPCD_KIT_DIR
#
#       MODIFIED    (MM/DD/YY)
#        mpoess          12/16/98 - update to version 8.1.6
#       mpoess          09/25/98 - update audit
#       akarasik        07/29/98 -
#       akarasik        07/29/98 - Creation
#

. $KIT_DIR/env

# May need to change the following:
RSH=rsh

OH=$ORACLE_HOME
#ACID_DIR=$KIT_DIR/acid is set in env
OUT_DIR=$ACID_OUT

TXN1FILE=$OUT_DIR/txn1$$.out
```

```
TXN2FILE=$OUT_DIR/txn2$$.out
KEYFILE=$OUT_DIR/key$$.out
ISOFILE=$OUT_DIR/iso1

USER=$DATABASE_USER
PROG=atranspl


/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE

trap "/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE; exit 1" 1 2 3 15

usage() {

   echo ""
    echo "Usage: $0 [-u user/passwd] [-n remote_node] -h"
   echo ""
   exit 1;
}

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

while :
do
   case "$1" in
    -u) shift; USER=$1;;
    -n) shift; HOST="$1";;
    -h) usage; exit 0;;
    --) break;;
    esac
    shift;
done



de=`direxists.sh $ACID_OUT c` # I am not using $de afterward, but I want to avoid the output of direxists

# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

OKEY=`cat $KEYFILE | awk '{print $1}'`
echo "o_key is "$OKEY

# before the ACID transaction, let's run a ACID query to record the
# initial state of lineitem

echo "Running ACID query BEFORE the start of Isolation Test 1" >> $TXN2FILE
echo "`date`" >> $TXN2FILE
echo "" >> $TXN2FILE
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >> $TXN2FILE
echo "" >> $TXN2FILE
echo "-------------------------------------------------" >> $TXN2FILE

sleep 1

# start ACID transaction, Sleep for 60 second before COMMIT

$PROG 1 1 1 0 i$KEYFILE u$USER s60 b0 >> $TXN1FILE &

# let's sleep 10 seconds before starting ACID query

sleep 15

# start ACID query with the same OKEY

echo "Running ACID query 15 seconds AFTER the start of ACID Transaction" \
>> $TXN2FILE
echo "`date`" >> $TXN2FILE
if [ "$HOST" != "" ]
then
echo "Starting ACID query on node $HOST" >> $TXN2FILE
${RSH} -n ${HOST} sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >>
```

```
$TXN2FILE
else
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >> $TXN2FILE
fi

echo "------------------------------------------------" >> $TXN2FILE
wait
echo "------------------------------------------------" >> $TXN1FILE

cat $TXN1FILE $TXN2FILE >> $ISOFILE

/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE
```

c.18 iso2.sh

```
#!/bin/ksh
#
# $Header: iso2.sh 04-aug-99.09:19:54 mpoess Exp $
#
# iso2.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      iso2.sh - <one-line expansion of the name>
#
#    DESCRIPTION
#      Usage: iso2.sh [-u user/password] [-n remote_node] -h
#      Options: See usage below
#    NOTES
#      For a cross node isolation test, assume the local node is
#      one of the participating nodes. The other node can be
#      specified by the -n option.
#      You need to set the environment variable TPCD_KIT_DIR
#
#    MODIFIED    (MM/DD/YY)
#    mpoess      08/04/99 - Creation
#    mpoess      08/04/99 - Creation
#
#
#
================================================================
+
# May need to change the following:

. $KIT_DIR/env

RSH=rsh

OH=$ORACLE_HOME
# ACID_DIR=$TPCD_KIT_DIR/audit is set in env
OUT_DIR=$ACID_OUT

DURA_DIR=$ACID_DIR/dura

TXN1FILE=$OUT_DIR/txn1$$.out
TXN2FILE=$OUT_DIR/txn2$$.out
KEYFILE=$OUT_DIR/key$$.out
ISOFILE=$OUT_DIR/iso2

USER=$DATABASE_USER
PROG=atranspl

/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE

trap "/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE; exit 1" 1 2 3 15
```

```
usage() {
    echo ""
     echo "Usage: $0 [-u user/passwd] [-n remote_node] -h"
    echo ""
    exit 1;
}

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

while :
do
   case "$1" in
    -u) shift; USER=$1;;
    -n) shift; HOST="$1";;
    -h) usage; exit 0;;
    --) break;;
    esac
    shift;
done



# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

OKEY=`cat $KEYFILE | awk '{print $1}'`
echo "o_key is "$OKEY

# before the ACID transaction, let's run a ACID query to record the
# initial state of lineitem

echo "Running ACID query BEFORE the start of Isolation Test 1" >> $TXN2FILE
echo "`date`" >> $TXN2FILE
echo "" >> $TXN2FILE
sqlplus "$USER" @$ACID_DIR/isolation/a_query $OKEY >> $TXN2FILE
echo "" >> $TXN2FILE
echo "------------------------------------------------" >> $TXN2FILE
sleep 1

# start ACID transaction, Sleep for 30 second before ROLLBACK

$PROG 1 1 0 0 i$KEYFILE u$USER s30 >> $TXN1FILE &

# let's sleep 15 seconds before starting ACID query

sleep 15

# start ACID query with the same OKEY

echo "Running ACID query 15 seconds AFTER the start of ACID transaction" \
>> $TXN2FILE
echo "`date`" >> $TXN2FILE
if [ "$HOST" != "" ]
then
echo "Starting ACID query on node $HOST" >> $TXN2FILE
${RSH} -n ${HOST} sqlplus "$USER" @$ACID_DIR/isolation/a_query $OKEY >> $TXN2FILE
else
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >> $TXN2FILE
fi

echo "------------------------------------------------" >> $TXN2FILE
wait
echo "------------------------------------------------" >> $TXN1FILE

cat $TXN1FILE $TXN2FILE >> $ISOFILE

/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE
```

c.19 iso3.sh

```
#!/bin/ksh
#
# $Header: iso3.sh 04-aug-99.09:20:35 mpoess Exp $
#
# iso3.sh
#
# Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#       NAME
#         iso3.sh - <one-line expansion of the name>
#
#   DESCRIPTION
#         Usage: iso3.sh [-u user/password] [-n remote_node] -h
#       Options: See usage below
#   NOTES
#             For a cross node isolation test, assume the local node is
#         one of the participating nodes. The other node can be
#           specified by the -n option.
#     We need to make sure the remote node has access to the
#     file system on the local node. Otherwise, we need to rcp
#       the keyfile to the remote system.
#             You need to set the environment variable TPCD_KIT_DIR
#
#       MODIFIED     (MM/DD/YY)
#       mpoess         08/04/99 - Creation
#       mpoess         08/04/99 - Creation
#

. $KIT_DIR/env

# May need to change the following:
RSH=rsh

OH=$ORACLE_HOME
#ACID_DIR=$TPCD_KIT_DIR/audit is set in env
OUT_DIR=$ACID_OUT

DURA_DIR=$ACID_DIR/dura

TXN1FILE=$OUT_DIR/txn1$$.out
TXN2FILE=$OUT_DIR/txn2$$.out
KEYFILE=$OUT_DIR/key$$.out
ISOFILE=$OUT_DIR/iso3

USER=$DATABASE_USER
PROG=atranspl

/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE

trap "/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE; exit 1" 1 2 3 15

usage() {

   echo ""
     echo "Usage: $0 [-u user/passwd] [-n remote_node] -h"
   echo ""
   exit 1;
}

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

while :
do
   case "$1" in
    -u) shift; USER=$1;;
    -n) shift; HOST="$1";;
    -h) usage; exit 0;;
    --) break;;
    esac
```

```
    shift
done


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE
rcp $KEYFILE ${HOST}:$KEYFILE

sleep 1

# start ACID transaction, Sleep for 30 second before COMMIT

$PROG 1 2 1 0 i$KEYFILE u$USER s30 b0 >> $TXN1FILE &

# let's sleep 15 seconds before starting second ACID transaction

sleep 15

# start another ACID transaction with the same LKEY and OKEY
# but different DELTA

# Do not sleep before COMMIT so that we can see TXN2 has waited.

if [ "$HOST" != "" ]
then
echo "Starting TXN2 on node $HOST" >> $TXN2FILE
${RSH} -n ${HOST} $PROG 2 2 1 1 i$KEYFILE u$USER s1 b1 >> $TXN2FILE &
else
$PROG 2 2 1 1 i$KEYFILE u$USER s1 b1 >> $TXN2FILE &
fi

wait
echo "------------------------------------------------" >> $TXN2FILE
echo "------------------------------------------------" >> $TXN1FILE

cat $TXN1FILE $TXN2FILE >> $ISOFILE

/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE
```

c.20 iso4.sh

```
#!/bin/ksh
#
# $Header: iso4.sh 04-aug-99.09:21:12 mpoess Exp $
#
# iso4.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#      NAME
#        iso4.sh - <one-line expansion of the name>
#
#   DESCRIPTION
#        Usage: iso4.sh [-u user/password] [-n remote_node] -h
#       Options: See usage below
#   NOTES
#            For a cross node isolation test, assume the local node is
#        one of the participating nodes. The other node can be
#        specified by the -n option.
#            We need to make sure the remote node has access to the
#            file system on the local node. Otherwise, we need to rcp
#        the keyfile to the remote system.
#            You need to set the environment variable TPCD_KIT_DIR
```

```
#
#       MODIFIED     (MM/DD/YY)
#       mpoess       08/04/99  - Creation
#       mpoess       08/04/99  - Creation
#

. $KIT_DIR/env

# May need to change the following:
RSH=rsh

OH=$ORACLE_HOME
# ACID_DIR=$TPCD_KIT_DIR/audit is set in
env
OUT_DIR=$ACID_OUT

DURA_DIR=$ACID_DIR/dura

TXN1FILE=$OUT_DIR/txn1$$.out
TXN2FILE=$OUT_DIR/txn2$$.out
KEYFILE=$OUT_DIR/key$$.out
ISOFILE=$OUT_DIR/iso4

USER=$DATABASE_USER
PROG=atranspl

/bin/rm -rf $TXN1FILE $TXN2FILE
$KEYFILE

trap "/bin/rm -rf $TXN1FILE $TXN2FILE
$KEYFILE; exit 1" 1 2 3 15

usage() {

   echo ""
     echo "Usage: $0 [-u user/passwd] [-n
remote_node] -h"
   echo ""
   exit 1;
}

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

while :
do
    case "$1" in
    -u) shift; USER=$1;;
    -n) shift; HOST="$1";;
    -h) usage; exit 0;;
    --) break;;
    esac
    shift
done


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE
rcp $KEYFILE ${HOST}:$KEYFILE

sleep 1

# start ACID transaction, Sleep for 30 second
before ROLLBACK

$PROG 1 2 0 0 i$KEYFILE u$USER s30 b0
>> $TXN1FILE &

# let's sleep 15 seconds before starting second
ACID transaction

sleep 15

# start another ACID transaction with the same
LKEY and OKEY
# but different DELTA

# Do not sleep before COMMIT so that we can
see TXN2 has waited.

if [ "$HOST" != "" ]
then
echo "Starting TXN2 on node $HOST" >>
$TXN2FILE
${RSH} -n ${HOST} $PROG 2 2 1 1
i$KEYFILE u$USER s1 b1 >> $TXN2FILE &
else
$PROG 2 2 1 1 i$KEYFILE u$USER s1 b1 >>
```

```
$TXN2FILE &
fi

wait
echo "------------------------------------------------"
>> $TXN2FILE
echo "------------------------------------------------"
>> $TXN1FILE

cat $TXN1FILE $TXN2FILE >> $ISOFILE

/bin/rm -rf $TXN1FILE $TXN2FILE
$KEYFILE
```

c.21 iso5.sh

```
#!/bin/ksh
#
# $Header: iso5.sh 04-aug-99.09:21:45 mpoess
Exp $
#
# iso5.sh
#
# Copyright (c) Oracle Corporation 1999. All
Rights Reserved.
#
#      NAME
#        iso5.sh - <one-line expansion of the
name>
#
#   DESCRIPTION
#        Usage: iso5.sh [-u user/password] [-n
remote_node] -h
#       Options: See usage below
#   NOTES
#             For a cross node isolation test,
assume the local node is
#        one of the participating nodes. The
other node can be
#        specified by the -n option.
#             You need to set the environment
variable TPCD_KIT_DIR
#
#      MODIFIED     (MM/DD/YY)
#      mpoess       08/04/99 - Creation
#      mpoess       08/04/99 - Creation
#

. $KIT_DIR/env

# May need to change the following:
RSH=rsh

OH=$ORACLE_HOME
# ACID_DIR=$TPCD_KIT_DIR/audit is set in
env
OUT_DIR=$ACID_OUT
DURA_DIR=$ACID_DIR/dura

TXN1FILE=$OUT_DIR/txn1$$.out
TXN2FILE=$OUT_DIR/txn2$$.out
KEYFILE=$OUT_DIR/key$$.out
ISOFILE=$OUT_DIR/iso5

USER=$DATABASE_USER
PROG=atranspl

/bin/rm -rf $TXN1FILE $TXN2FILE
$KEYFILE

trap "/bin/rm -rf $TXN1FILE $TXN2FILE
$KEYFILE; exit 1" 1 2 3 15

usage() {

   echo ""
     echo "Usage: $0 [-u user/passwd] [-n
remote_node] -h"
   echo ""
   exit 1;
}

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

```
while :
do
   case "$1" in
    -u) shift; USER=$1;;
    -n) shift; HOST="$1";;
    -h) usage; exit 0;;
    --) break;;
    esac
    shift;
done



# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE
rcp $KEYFILE ${HOST}:$KEYFILE

OKEY=`cat $KEYFILE | awk '{print $1}'`
echo "o_key is "$OKEY

# before the ACID transaction, let's run a ACID query to record the
# initial state of lineitem

echo "Running ACID query BEFORE the start of Isolation Test 5" >> $TXN1FILE
echo "`date`" >> $TXN1FILE
echo "" >> $TXN1FILE
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >> $TXN1FILE
echo "" >> $TXN1FILE
echo "-----------------------------------------------" >> $TXN1FILE

sleep 1

# start ACID transaction, Sleep for 60 second before COMMIT

$PROG 1 1 1 0 i$KEYFILE u$USER s60 >> $TXN1FILE &

# let's sleep 5 seconds before starting PARTSUPP query

sleep 5

# First generate PS_PARTKEY and PS_SUPPKEY

PSKEY=`randpsup 1`

echo "Running PARTSUPP query 5 seconds AFTER the start of ACID Transaction" \
>> $TXN2FILE
echo "`date`" >> $TXN2FILE
echo "PS_PARTKEY and PS_SUPPKEY are: $PSKEY" >> $TXN2FILE

if [ "$HOST" != "" ]
then
echo "Starting PARTSUPP query on node $HOST" >> $TXN2FILE
${RSH} -n ${HOST} sqlplus $USER @$ACID_DIR/isolation/a_query2 ${PSKEY} >> $TXN2FILE &
else
sqlplus $USER @$ACID_DIR/isolation/a_query2 ${PSKEY} >> $TXN2FILE &
fi

wait

echo "-----------------------------------------------" >> $TXN2FILE
echo "-----------------------------------------------" >> $TXN1FILE

cat $TXN1FILE $TXN2FILE >> $ISOFILE

/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE
```

c.22 iso6.sh

```
#!/bin/ksh
#
# $Header: iso6.sh 04-aug-99.09:22:12 mpoess Exp $
#
# iso6.sh
#
# Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#       NAME
#         iso6.sh - <one-line expansion of the name>
#
#   DESCRIPTION
#         Usage: iso6.sh [-u user/password] [-n remote_node] -h
#         Options: See usage below
#   NOTES
#             For a cross node isolation test, assume the local node is
#         one of the participating nodes. The other node can be
#         specified by the -n option.
#           We need to make sure the remote node has access to the
#             file system on the local node. Otherwise, we need to rcp
#         the keyfile to the remote system.
#             You need to set the environment variable TPCD_KIT_DIR
#
#       MODIFIED     (MM/DD/YY)
#       mpoess          08/04/99  - Creation
#       mpoess          08/04/99 - Creation
#

. $KIT_DIR/env

# May need to change the following:
RSH=rsh
OH=/private/tpcd
# ACID_DIR=$TPCD_KIT_DIR/audit is set in env
OUT_DIR=$ACID_OUT

DURA_DIR=$ACID_DIR/dura

TXN1FILE=$OUT_DIR/txn1$$.out
TXN2FILE=$OUT_DIR/txn2$$.out
TXN3FILE=$OUT_DIR/txn3$$.out
KEYFILE=$OUT_DIR/key$$.out
ISOFILE=$OUT_DIR/iso6

USER=$DATABASE_USER
PROG=atranspl

/bin/rm -rf $TXN1FILE $TXN2FILE $TXN3FILE $KEYFILE

trap "/bin/rm -rf $TXN1FILE $TXN2FILE $TXN3FILE $KEYFILE; exit 1" 1 2 3 15

usage() {

    echo ""
      echo "Usage: $0 [-u user/passwd] [-n remote_node] -h"
    echo ""
    exit 1;
}

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

while :
do
    case "$1" in
     -u) shift; USER=$1;;
     -n) shift; HOST="$1";;
     -h) usage; exit 0;;
     --) break;;
     esac
     shift;
done
```

```
# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE
rcp $KEYFILE ${HOST}:$KEYFILE

OKEY=`cat $KEYFILE | awk '{print $1}'`
echo "o_key is "$OKEY

# before the any transaction, let's run a ACID query to record the
# initial state of lineitem

echo "Running ACID query BEFORE the start of Isolation Test 6" >> $TXN2FILE
echo "`date`" >> $TXN2FILE
echo "" >> $TXN2FILE
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >> $TXN2FILE
echo "" >> $TXN2FILE
echo "------------------------------------------------" >> $TXN2FILE

sleep 1

# start Query 1, use 0 as the delta

echo "Running Query 17b at `date`" >> $TXN1FILE
sqlplus $USER @q1 >> $TXN1FILE &

# sleep 2 seconds before starting ACID transaction

sleep 2

# start ACID transaction, COMMIT after one second

echo "Starting AICD transaction at `date`" >> $TXN2FILE

if [ "$HOST" != "" ]
then
echo "Starting ACID transaction on node $HOST" >> $TXN2FILE
${RSH} -n ${HOST} $PROG 1 1 1 0 i$KEYFILE u$USER s1 >> $TXN2FILE &
else
$PROG 1 1 1 0 i$KEYFILE u$USER s1 >> $TXN2FILE &
fi

# start Query 17

sleep 2

echo "Running 2nd Query 17b at `date`" >> $TXN3FILE
sqlplus $USER @q1 >> $TXN3FILE &
# wait for everyone to finish

wait

echo "------------------------------------------------" >> $TXN3FILE
echo "------------------------------------------------" >> $TXN2FILE
echo "------------------------------------------------" >> $TXN1FILE

cat $TXN1FILE $TXN2FILE $TXN3FILE >> $ISOFILE

/bin/rm -rf $TXN1FILE $TXN2FILE $TXN3FILE $KEYFILE
```

c.23 prepare4acid.sh

```
#!/bin/ksh
#
#           $Header:           prepare4acid.sh
```

```
12-aug-99.17:09:18 mpoess Exp $
#
# prepare4acid.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#      NAME
#        prepare4acid.sh
#
#      DESCRIPTION
#       Prepares the qualification database for the acid tests.
#
#      NOTES
#
#      MODIFIED     (MM/DD/YY)
#      mpoess         08/12/99  - Creation
#      mpoess         08/12/99  - Creation
#
. $KIT_DIR/env

sqlplus $DATABASE_USER @d_hist
sqlplus $DATABASE_USER @atrans
```

c.24 randkey.c

```
/* Copyright (c) 2001, 2002, Oracle Corporation.  All rights reserved.  */

/*

   NAME
     randkey.c  - <one-line expansion of the name>

   DESCRIPTION
     Generate random keys for ACID transactions:
     O_ORDERKEY unique random (1..SF*150000*4) and only
          first 8 keys out of every 32 are populated.
          and
     L_ORDERKEY based on Clause 3.1.6.2
     DELTA random (1..100)
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "atranspl.h"

#define ORDERCNT 150000.0

/* MK_SPARSE adopted from dss.h */

#define MK_SPARSE(key, seq) \
          (((((key>>3)<<2)|(seq & 0x0003))<<3)|(key & 0x0007))

void sql_error();
void usage();
void ACIDinit();
long atol();
void srand48();
long lrand48();

/* Not really used here, but retained it for future purposes. */

typedef struct aciddef {
  long okey;
  long lkey;
  int delta;
} adef;

long l_key = 0;
long o_key = 0;
char lname[UNAME_LEN];
char *passwd;

/* OCI handles */
```

```
OCIEnv *tpcenv;
OCIServer *tpcsrv;
OCIError *errhp;
OCISvcCtx *tpcsvc;
OCISession *tpcusr;
OCIStmt *curi;

OCIBind *l_key_bp;
OCIBind *o_key_bp;

sword status = OCI_SUCCESS;  /* OCI return value */

char sqlstmt[1024];


void ACIDexit() {
  OCILogoff(tpcsvc,errhp);
  OCIhfree(tpcenv,OCI_HTYPE_STMT);
  OCIhfree(tpcsvc,OCI_HTYPE_SVCCTX);
  OCIhfree(tpcsrv,OCI_HTYPE_SERVER);
  OCIhfree(tpcusr,OCI_HTYPE_SESSION);
}


/* type: 0 if environment handle is passed, 1 if error handle is passwd */

void sql_error(errhp,status,type)
      OCIError *errhp;
      sword status;
      sword type;
{
  char msg[2048];
  sb4 errcode;
  ub4 msglen;
  int i,j;

  switch(status) {
  case OCI_SUCCESS_WITH_INFO:
    fprintf(stderr, "Error: Statement returned with info.\n");
    if (type)
      (void) OCIErrorGet(errhp,1,NULL,(sb4 *) &errcode,(text *)msg,
2048,OCI_HTYPE_ERROR);
    else
      (void) OCIErrorGet(errhp,1,NULL,(sb4 *) &errcode,(text *)msg,
2048,OCI_HTYPE_ENV);
    fprintf(stderr,"%s\n",msg);
    break;
  case OCI_ERROR:
    fprintf(stderr, "Error: OCI call error.\n");
    if (type)
      (void) OCIErrorGet(errhp,1,NULL,(sb4 *) &errcode,(text *)msg,
2048,OCI_HTYPE_ERROR);
    else
      (void) OCIErrorGet(errhp,1,NULL,(sb4 *) &errcode,(text *)msg,
2048,OCI_HTYPE_ENV);
    fprintf(stderr,"%s\n",msg);
    break;
  case OCI_INVALID_HANDLE:
    fprintf(stderr, "Error: Invalid Handle.\n");
    if (type)
      (void) OCIErrorGet(errhp,1,NULL,(sb4 *) &errcode,(text *)msg,
2048,OCI_HTYPE_ERROR);
    else
      (void) OCIErrorGet(errhp,1,NULL,(sb4 *) &errcode,(text *)msg,
2048,OCI_HTYPE_ENV);
    fprintf(stderr,"%s\n",msg);
    break;
```

```
  }
 /* Rollback just in case */

  (void)
OCITransRollback(tpcsvc,errhp,OCI_DEFAU
LT);

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

  ACIDexit();

  exit(1);
}


main(argc, argv)
      int argc;
      char **argv;
{

  long count;
  long i;
  double sf;            /* need to accomodate
sf 0.1 */
  double random;
  double ordcnt;
  adef *res;

  if ((argc < 3) || (argc > 4)) {
     usage();
     exit(-1);
  }

 strcpy((char *) lname, "tpcd/tpcd");

  count = atol(argv[1]);
  sf = atof(argv[2]);

  argc -= 2;
  argv += 2;

  while (--argc) {
     ++argv;
     switch(argv[0][0]) {
     case 'u':
        strncpy((char *) lname, ++(argv[0]),
UNAME_LEN);
       if (strchr((char *) lname, '/') == NULL)
{
          usage();
          exit(-1);
        }
        break;
      default:
        fprintf(stderr, "Unknown argument
%s\n", argv[0]);
        usage();
        break;
      }
  }

  ACIDinit();

 /* initialize array for random numbers */

 res = (adef *) malloc(count*sizeof(adef));
  ordcnt = (double) ORDERCNT * (double)
sf;

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

      /*              The              algorithm:
*/
   /* Assumes drand's output is 'unique', first
get a number within */
     /* the range of [0..sf*ORDERCNT) and
then maps the different      */
      /* ranges to generate the real output.
*/

      random = floor(drand48() * (double)
ordcnt) + 1;
      res[i].okey = o_key = (long)
MK_SPARSE((long) random, 0);
    res[i].delta = (long) floor(drand48() * 100)
+ 1;
```

```
    /* Obtain l_key from l_key query */

      OCIsexec(tpcsvc,curi,errhp,1);

       /* l_key is the highest l_linenumber
available.  We need to pick */
      /* at random a number between 1..l_key.
*/

    res[i].lkey = (lrand48() % l_key) + 1;

      printf("%ld  %ld  %d\n",  res[i].okey,
res[i].lkey, res[i].delta);
   }

   ACIDexit();
   free(res);

}


void usage() {

   fprintf(stderr, "Usage: randkey <number of
random  keys  to  generate>  <SF>
u<user/password>\n");
   fprintf(stderr, "\n");
}

void ACIDinit()
{

  /* run random seed */

   srand48(getpid());

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

   (void)  OCIInitialize(OCI_DEFAULT,(dvoid
*)0,0,0,0);
   if((status=OCIEnvInit((OCIEnv
**)&tpcenv,OCI_DEFAULT,0,(dvoid
**)0)) !=
         OCI_SUCCESS)
      sql_error(tpcenv, status, 0);


OCIhalloc(tpcenv,&errhp,OCI_HTYPE_ERR
OR);

OCIhalloc(tpcenv,&curi,OCI_HTYPE_STMT)
;

OCIhalloc(tpcenv,&tpcsvc,OCI_HTYPE_SVC
CTX);

OCIhalloc(tpcenv,&tpcsrv,OCI_HTYPE_SER
VER);

OCIhalloc(tpcenv,&tpcusr,OCI_HTYPE_SESS
ION);

  /* get username and password */

  passwd = strchr(lname, '/');
  *passwd = '\0';
   passwd++;

   if
((status=OCIServerAttach(tpcsrv,errhp,(text
*)0,0,OCI_DEFAULT))!=OCI_SUCCESS)
       sql_error(errhp,status,1);


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcsrv
,0,OCI_ATTR_SERVER,errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,lname
,strlen(lname),OCI_ATTR_USERNAME,
              errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,passw
d,strlen(passwd),OCI_ATTR_PASSWORD,
              errhp);
```

```
  if ((status = OCISessionBegin(tpcsvc, errhp,
tpcusr, OCI_CRED_RDBMS,

OCI_DEFAULT)) != OCI_SUCCESS)
      sql_error(errhp,status,1);


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcusr
,0,OCI_ATTR_SESSION,errhp);

   /* Open and Parse cursor for query to
choose determine l_key. */
  /*  Binds  l_key  to  :l_key.
*/

  sprintf((char *) sqlstmt,SQLTXT1);
   OCIStmtPrepare(curi,errhp,(text
*)sqlstmt,strlen((char *)sqlstmt),

OCI_NTV_SYNTAX,OCI_DEFAULT);


OCIbbname(curi,l_key_bp,errhp,":l_key",AD
R(l_key),SIZ(l_key),SQLT_INT);

OCIbbname(curi,o_key_bp,errhp,":o_key",AD
R(o_key),SIZ(o_key),SQLT_INT);
}
```

c.25 randpsup.c

```
/* Copyright (c) 2001, 2002, Oracle
Corporation.  All rights reserved.  */

/*

   NAME
     randpsup.c  - <one-line expansion of the
name>

   DESCRIPTION
     Generate random keys for ACID
PARTSUPP transactions:
     (Clause 4.2.3)
      PS_PARTKEY random within
[SF*200000]
          and
     PS_SUPPKEY = (PS_PARTKEY + (i *
((S/4) + (int)(PS_PARTKEY - 1)
                  /S))) % S + 1
      where i random within [0..3] and S = SF
* 10000

MODIFIED
    mpoess                10/23/02 -
mpoess_update_from_visa
   mpoess        01/04/01 - Creation

*/

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

#define PS_PER_SF 200000.0
#define S_PER_SF 10000.0
#define SUPP_PER_PART 4

/* borrowed from build.c in the dbgen
distribution */

#define PART_SUPP_BRIDGE(tgt, p, s) \
    { \
   long tot_scnt = (long) (S_PER_SF * sf); \
     tgt  =  (p  +  s  *  (tot_scnt  /
SUPP_PER_PART +  \
          (long)  ((p  - 1) / tot_scnt))) % tot_scnt
+ 1; \
     }

void usage();
double atof();
void srand48();
long lrand48();
```

```
}


main(argc, argv)
      int argc;
      char **argv;
{

  double sf = 0.1;              /* scale factor
*/
  long supp;                    /* the ith
supplier */
  long pkey;                    /* partkey
*/
  long maxpkey;                 /* highest
partkey    */
    long ps_skey;                       /*
ps_suppkey          */

 if (argc < 2) {
     usage();
     exit(-1);
  }

 /* seed the random number generator */

  srand48(getpid());

 sf = atof(argv[1]);
 maxpkey = (long) (sf * PS_PER_SF);
 supp = lrand48() % 4;
 pkey = lrand48() % maxpkey + 1;

  PART_SUPP_BRIDGE(ps_skey,       pkey,
supp);

 fprintf(stdout, "%ld %ld", pkey, ps_skey);

  exit(0);
}


void usage()
{
  fprintf(stderr, "Usage: randpsup <SF>\n\n");
```

c.26 run_acid.sh

```
#!/bin/ksh
#
# $Header: run_acid.sh 08-aug-99.15:30:10
mpoess Exp $
#
# run_acid.sh
#
# Copyright (c) Oracle Corporation 1999. All
Rights Reserved.
#
#      NAME
#        run_acid.sh - <one-line expansion of
the name>
#
#   DESCRIPTION
#        Usage: run_acid.sh [-n iter] [-s stream]
[-p prog] [-i infile]
#                                [-o outfile] [-d
durafile] [-u usr/pswd]
#                                [-t trigger] [-f
scale factor] -h
#
#        Options: See usage below
#
#      MODIFIED     (MM/DD/YY)
#      mpoess       08/08/99 - Creation
#      mpoess       08/08/99 - Creation
#

. $KIT_DIR/env

OH=$ORACLE_HOME
ACID_DIR=$ACID_DIR
OUT_DIR=$ACID_OUT
```

```
usage() {

   echo ""
    echo "Usage: $0 [-n iter] [-s stream] [-p prog] [-i infile] [-o outfile]"
     echo "                [-d durafile] [-u usr/pswd] -h"
   echo ""
    echo "-n iter       : number of iterations, default is 100"
    echo "-s stream     : number of streams, default is 2"
    echo "-p prog       : program to run, default is atranspl.ott"
    echo "-i infile     : input file prefix, suffix by process number within a"
    echo "                stream and run ID, default is ./acid_in"
    echo "-o outfile    : output file prefix, similar to input file"
    echo "                default is ./out/acid_out"
    echo "-d durafile : durability file prefix, used for durability tests"
    echo "                default is ./dura/acid_dura"
    echo "-u usr/pswd : user/password combo for database access, default is tpch/tpch"
    echo "-t trigger    : trigger time between process starts, default is 1 second"
    echo "-h            : print this usage summary"
   exit 1;
}

ITER=200
STEM=${NUM_STREAMS}
let STEM="$STEM + 1" # add one for the update stream
SF=1
PROG=atranspl
IN=${ACID_DIR}/acid_in
DURA_DIR=$ACID_OUT/dura
OUT=$DURA_DIR/drate
DURA=$DURA_DIR/dura
KEY=${DURA_DIR}/key$$_
echo "$$" > ${DURA_DIR}/shellpid
USER=tpch/tpch
TRIG=1
HCNT=duracntb

set -- `getopt "n:s:p:i:o:d:u:ht:f:" "$@"` || usage

# get all the options

while :
do
   case "$1" in
    -n) shift; ITER=$1;;
    -s) shift; STEM=$1;;
    -p) shift; PROG=$1;;
    -i) shift; IN=$1;;
    -o) shift; OUT=$1;;
    -d) shift; DURA=$1;;
    -u) shift; USER=$1;;
    -h) usage; exit 0;;
    -t) shift; TRIG=$1;;
    -f) shift; SF=$1;;
    --) break;;
    esac
    shift;
done


echo "Starting ACID run..."

i=0
T=`expr $STEM \* $TRIG + 6`

# Get history count before the run

sqlplus $USER @cnt_hist > $DURA_DIR/$HCNT 2>&1

while [ $i -lt $STEM ]
do
    randkey $ITER ${SF} u${USER} >
```

```
${KEY}${i} &
   i=`expr $i + 1`
done

wait
# perform the consistency

i=0
while [ $i -lt $STEM ]
do
    for j in `head -10 ${KEY}${i} | awk '{printf "%d ",$1}'`
    do
       sqlplus tpch/tpch @consist $j >> $DURA_DIR/duraconsb
    done
   i=`expr $i + 1`
done

echo "Starting Transaction Counting Program"
count_tx.sh $STEM 100 $DURA_DIR &

i=0
while [ $i -lt $STEM ]
do

     $PROG $i $STEM 1 0 i${KEY}${i} o${OUT}${i} d${DURA}${i} u$USER s1 &
   T=`expr $T - $TRIG`
   i=`expr $i + 1`

done

wait

echo "ACID run completed"
```

c.27 sample.sh

```
#!/bin/ksh
#
# $Header: sample.sh 08-aug-99.17:10:00 mpoess Exp $
#
# sample.sh
#
# Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      sample.sh - <one-line expansion of the name>
#
#    DESCRIPTION
#      <short description of component this file declares/defines>
#
#    NOTES
#                <other useful comments, qualifications, etc.>
#
#    MODIFIED    (MM/DD/YY)
#    mpoess      08/08/99 - Creation
#    mpoess      08/08/99 - Creation
#

# $1 durability output file

. $KIT_DIR/env

cat $1 | grep o_key | awk '{printf "%d \n", $2}' | head -106 > /tmp/okey$$
cat $1 | grep l_key | awk '{printf "%d \n", $2}' | head -106 > /tmp/lkey$$

paste /tmp/okey$$ /tmp/lkey$$ > /tmp/keys$$
tail -6 /tmp/keys$$ > /tmp/6keys$$

echo "Keys chosen are:"
cat /tmp/6keys$$

i=1
```

```
while [ $i -le 6 ]
do

j=`cat /tmp/6keys$$ | tail -${i} | head -1`
sqlplus tpch/tpch @sample $j
i=`expr $i + 1`
done

#/bin/rm -f /tmp/*key*
```

c.28 sample.sql

```
Rem
Rem $Header: sample.sql 08-aug-99.17:10:34 mpoess Exp $
Rem
Rem sample.sql
Rem
Rem Copyright (c) Oracle Corporation 1999. All Rights Reserved.
Rem
Rem NAME
Rem sample.sql - <one-line expansion of the name>
Rem
Rem DESCRIPTION
Rem <short description of component this file declares/defines>
Rem
Rem NOTES
Rem <other useful comments, qualifications, etc.>
Rem
Rem MODIFIED (MM/DD/YY)
Rem mpoess 08/08/99 - Creation
Rem mpoess 08/08/99 - Created
Rem

alter session set nls_date_format = 'YYYY-MM-DD HH:MI:SS';
select * from history where h_o_key = &&1 and h_l_key = &&2;

exit;
```

Appendix D Query text and Output

D.1 1.log

```
Begin Execution at Fri Oct 24 05:53:53 2003

-- using default substitutions

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') - 90
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 37734107.00
56586554400.73
53758257134.87 55909065222.83
25.52
38273.13 0.05
1478493.00
N F 991417.00
1487504710.38
1413082168.05 1469649223.19
25.52
38284.47 0.05
38854.00
N O 74476040.00
111701729697.74
106118230307.61 110367043872.50
25.50
38249.12 0.05
2920374.00
R F 37719753.00
56568041380.90
53741292684.60 55889619119.83
25.51
38250.85 0.05
1478870.00

4 rows processed.
Query Processed in 11.30 seconds.

Ended Executing this Stream at Fri Oct 24 05:54:04 2003

Stream Started at 1066946033.33
Stream Ended at 1066946044.63
```

```
Stream Processed in 11.30 seconds

SQL statements processed: 1
```

D.2 2.log

```
Begin Execution at Fri Oct 24 05:54:04 2003

-- using default substitutions

select * from (
select
s_acctbal,
s_name,
n_name,
p_partkey,
p_mfgr,
s_address,
s_phone,
s_comment
from
part,
supplier,
partsupp,
nation,
region
where
p_partkey = ps_partkey
and s_suppkey = ps_suppkey
and p_size = 15
and p_type like '%BRASS'
and s_nationkey = n_nationkey
and n_regionkey = r_regionkey
and r_name = 'EUROPE'
and ps_supplycost = (
select
min(ps_supplycost)
from
partsupp,
supplier,
nation,
region
where
p_partkey = ps_partkey
and s_suppkey = ps_suppkey
and s_nationkey = n_nationkey
and n_regionkey = r_regionkey
and r_name = 'EUROPE'
)
order by
s_acctbal desc,
n_name,
s_name,
p_partkey
)
where rownum <= 100

S_ACCTBAL                S_NAME
N_NAME
P_PARTKEY                P_MFGR
S_ADDRESS
S_PHONE
S_COMMENT
9938.53                  Supplier#000005359
UNITED KINGDOM
185358.00                Manufacturer#4
QKuHYh,vZGiwu2FWEJoLDx04
33-429-790-6131
blithely silent pinto beans are furiously. slyly
final deposits acros
9937.84                  Supplier#000005969
ROMANIA
108438.00                Manufacturer#1
ANDENSOSmk,miq23Xfb5RWt6dvUcvt6Qa
29-520-692-3537
carefully slow deposits use furiously. slyly ironic
platelets above the ironic
9936.22                  Supplier#000005250
UNITED KINGDOM
249.00                   Manufacturer#4
```

B3rqp0xbSEim4Mpy2RH J
33-320-228-2957
blithely special packages are. stealthily express deposits across the closely final instructi
9923.77 Supplier#000002324
GERMANY
29821.00 Manufacturer#4
y3OD9UywSTOk
17-779-299-1839
quickly express packages breach quiet pinto beans. requ
9871.22 Supplier#000006373
GERMANY
43868.00 Manufacturer#5
J8fcXWsTqM
17-813-485-8637
never silent deposits integrate furiously blit
9870.78 Supplier#000001286
GERMANY
81285.00 Manufacturer#2
YKA,E2fjiVd7eUrzp2Ef8j1QxGo2DFnosaTEH
17-516-924-4574
final theodolites cajole slyly special,
9870.78 Supplier#000001286
GERMANY
181285.00 Manufacturer#4
YKA,E2fjiVd7eUrzp2Ef8j1QxGo2DFnosaTEH
17-516-924-4574
final theodolites cajole slyly special,
9852.52 Supplier#000008973
RUSSIA
18972.00 Manufacturer#2
t5L67YdBYYH6o,Vz24jpDyQ9
32-188-594-7038
quickly regular instructions wake-- carefully unusual braids into the expres
9847.83 Supplier#000008097
RUSSIA
130557.00 Manufacturer#2
xMe97bpE69NzdwLoX
32-375-640-3593
slyly regular dependencies sleep slyly furiously express dep
9847.57 Supplier#000006345
FRANCE
86344.00 Manufacturer#1
VSt3rzk3qG698u6ld8HhOByvrTcSTSvQlDQDa g 16-886-766-7945
silent pinto beans should have to snooze carefully along the final reques
9847.57 Supplier#000006345
FRANCE
173827.00 Manufacturer#2
VSt3rzk3qG698u6ld8HhOByvrTcSTSvQlDQDa g 16-886-766-7945
silent pinto beans should have to snooze carefully along the final reques
9836.93 Supplier#000007342
RUSSIA
4841.00 Manufacturer#4
JOlK7C1,7xrEZSSOw
32-399-414-5385
final accounts haggle. bold accounts are furiously dugouts. furiously silent asymptotes are slyly
9817.10 Supplier#000002352
RUSSIA
124815.00 Manufacturer#2
4LfoHUZjgjEbAKw TgdKcgOc4D4uCYw
32-551-831-1437
blithely pending packages across the ironic accounts grow slyly after the furiously
9817.10 Supplier#000002352
RUSSIA
152351.00 Manufacturer#3
4LfoHUZjgjEbAKw TgdKcgOc4D4uCYw
32-551-831-1437
blithely pending packages across the ironic accounts grow slyly after the furiously
9739.86 Supplier#000003384
FRANCE
138357.00 Manufacturer#2
o,Z3v4POifevE k9U1b 6J1ucX,I
16-494-913-5925
slyly ironic theodolites hag
9721.95 Supplier#000008757
UNITED KINGDOM
156241.00 Manufacturer#3

Atg6GnM4dT2
33-821-407-2995
ironic, even dolphins above the furiously ironic foxes sleep slyly around the caref
9681.33 Supplier#000008406
RUSSIA
78405.00 Manufacturer#1
,qUuXcftUl
32-139-873-8571
furiously even deposits affix thinly special theodolites. furiou
9643.55 Supplier#000005148
ROMANIA
107617.00 Manufacturer#1
kT4ciVFslx9z4s79p Js825
29-252-617-4850
doggedly even ideas boost furiously against the furiously express
9624.82 Supplier#000001816
FRANCE
34306.00 Manufacturer#3
e7vab91vLJPWxxZnewmnDBpDmxYHrb
16-392-237-6726
blithely regular accounts cajole furiously. regular
9624.78 Supplier#000009658
ROMANIA
189657.00 Manufacturer#1
oE9uBgEfSS4opIcepXyAYM,x
29-748-876-2014
regular deposits haggle. furiously express asympto
9612.94 Supplier#000003228
ROMANIA
120715.00 Manufacturer#2
KDdpNKN3cWu7ZSrbdqp7AfSLxx,qWB
29-325-784-8187
carefully pending accounts serve. furiously close deposits boost slyly. q
9612.94 Supplier#000003228
ROMANIA
198189.00 Manufacturer#4
KDdpNKN3cWu7ZSrbdqp7AfSLxx,qWB
29-325-784-8187
carefully pending accounts serve. furiously close deposits boost slyly. q
9571.83 Supplier#000004305
ROMANIA
179270.00 Manufacturer#2
qNHZ7WmCzygwMPRDO9Ps
29-973-481-1831
furiously final deposits
9558.10 Supplier#000003532
UNITED KINGDOM
88515.00 Manufacturer#4
EOeuiiOn21OVpTlGguufFDFsbN1p0lhpxHp
33-152-301-2164
daring, sly accounts breach about th
9492.79 Supplier#000005975
GERMANY
25974.00 Manufacturer#5
S6mIiCTx82z7lV
17-992-579-4839
always pending packages boost slyly.
9461.05 Supplier#000002536
UNITED KINGDOM
20033.00 Manufacturer#1
8mmGbyzaU 7ZS2wJumTibypncu9pNkDc4FYA
33-556-973-5522
even foxes are quickly furiously express requests. packages
9453.01 Supplier#000000802
ROMANIA
175767.00 Manufacturer#1
,6HYXb4uaHITmtMBj4Ak57Pd
29-342-882-6463
final, regular packages across the slowly regular packag
9408.65 Supplier#000007772
UNITED KINGDOM
117771.00 Manufacturer#4
AiC5YAH,gdu0i7
33-152-491-1126
blithely final ideas sleep carefully. requests are
9359.61 Supplier#000004856
ROMANIA
62349.00 Manufacturer#5
HYogcF3Jb yh1
29-334-870-9731

carefully unusual packages sleep carefully even ideas. dogged accoun
9357.45 Supplier#000006188 UNITED KINGDOM
138648.00 Manufacturer#1 g801,ssP8wpTk4Hm
33-583-607-1633
carefully regular deposits wake carefully furiously even i
9352.04 Supplier#000003439 GERMANY
170921.00 Manufacturer#4 qYPDgoiBGhCYxjgC
17-128-996-4650
fluffily regular pinto beans wake. unusual, final ideas c
9312.97 Supplier#000007807 RUSSIA
90279.00 Manufacturer#5 oGYMPCk9XHGB2PBfKRnHA
32-673-872-5854
unusual asymptotes above the
9312.97 Supplier#000007807 RUSSIA
100276.00 Manufacturer#5 oGYMPCk9XHGB2PBfKRnHA
32-673-872-5854
unusual asymptotes above the
9280.27 Supplier#000007194 ROMANIA
47193.00 Manufacturer#3 zhRUQkBSrFYxIAXTfInj vyGRQjeK
29-318-454-2133
slyly ironic requests despite the unusual ins
9274.80 Supplier#000008854 RUSSIA
76346.00 Manufacturer#3 1xhLoOUM7I3mZ1mKnerw OSqdbb4QbGa
32-524-148-5221
ruthlessly ironic instructions along the regular, furious requests integrate car
9249.35 Supplier#000003973 FRANCE
26466.00 Manufacturer#1 d18GiDsL6Wm2IsGXM,RZf1jCsgZAOjNYVT hTRP4 16-722-866-1658
quickly ironic sauternes use b
9249.35 Supplier#000003973 FRANCE
33972.00 Manufacturer#1 d18GiDsL6Wm2IsGXM,RZf1jCsgZAOjNYVT hTRP4 16-722-866-1658
quickly ironic sauternes use b
9208.70 Supplier#000007769 ROMANIA
40256.00 Manufacturer#5 rsimdze 5o9P Ht7xS
29-964-424-9649
furiously ruthless epitaphs among the furiously regular accounts use slowly fluffily ev
9201.47 Supplier#000009690 UNITED KINGDOM
67183.00 Manufacturer#5 CB BnUTlmi5zdeEl7R7
33-121-267-9529
blithely unusual accounts integrate slyly. platelets
9192.10 Supplier#000000115 UNITED KINGDOM
85098.00 Manufacturer#3 nJ 2t0f7Ve,wL1,6WzGBJLNBUCKlsV
33-597-248-1220
slyly bold pinto beans boost across the furiously regular packages. carefully regu
9189.98 Supplier#000001226 GERMANY
21225.00 Manufacturer#4 qsLCqSvLyZfuXIpjz
17-725-903-1381
final, express instruction
9128.97 Supplier#000004311 RUSSIA
146768.00 Manufacturer#5 I8IjnXd7NSJRs594RxsRR0
32-155-440-7120
regular pinto beans sleep ca
9104.83 Supplier#000008520 GERMANY

150974.00 Manufacturer#4 RqRVDgD0ER J9 b41vR2,3 17-728-804-1793 deposits sleep carefully e
9101.00 Supplier#000005791 ROMANIA
128254.00 Manufacturer#5 zub2zCV,jhHPPQqi,P2INAjE1zI n66cOEoXFG 29-549-251-5384 carefully ironic packages after the
9094.57 Supplier#000004582 RUSSIA
39575.00 Manufacturer#1 WB0XkCSG3r,mnQ n,h9VIxjjr9ARHFvKgMDf 32-587-577-1351 asymptotes above the slyly even requests haggle furiously about the regular accounts
8996.87 Supplier#000004702 FRANCE
102191.00 Manufacturer#5 8XVcQK23akp 16-811-269-8946 stealthy requests haggle c
8996.14 Supplier#000009814 ROMANIA
139813.00 Manufacturer#2 af0O5pg83lPU4IDVmEylXZVqYZQzSDlYLA mR 29-995-571-8781 ironic theodolites are evenly unusual requests -- pending pinto beans across the in
8968.42 Supplier#000010000 ROMANIA
119999.00 Manufacturer#5 aTGLEusCiL4F PDBdv665XBJhPyCOB0i 29-578-432-2146 furiously final ideas believe furiously. furiously final ideas
8936.82 Supplier#000007043 UNITED KINGDOM
109512.00 Manufacturer#1 FVajceZInZdbJE6Z9XsRUxrUEpiwHDrOXi,1R z 33-784-177-8208 furiously regular excuses wake after the blithely special pinto beans? even instructions sl
8929.42 Supplier#000008770 FRANCE
173735.00 Manufacturer#4 R7cG26TtXrHAP9 HckhfRi 16-242-746-9248 final accounts sleep furiously. blithely ironic foxes wake boldly across the furiously s
8920.59 Supplier#000003967 ROMANIA
26460.00 Manufacturer#1 eHoAXe62SY9 29-194-731-3944 quickly even requests should have to affix blithely -- fur
8920.59 Supplier#000003967 ROMANIA
173966.00 Manufacturer#2 eHoAXe62SY9 29-194-731-3944 quickly even requests should have to affix blithely -- fur
8913.96 Supplier#000004603 UNITED KINGDOM
137063.00 Manufacturer#2 OUzlvMUr7n,utLxmPNeYKSf3T24OXskxB5 33-789-255-7342 slyly ironic packages detect furious accounts. ironic de
8877.82 Supplier#000007967 FRANCE
167966.00 Manufacturer#5 A3pi1BARM4nx6R,qrwFoRPU 16-442-147-9345 final deposits after the silent deposits ha
8862.24 Supplier#000003323 ROMANIA
73322.00 Manufacturer#3 W9 lYcsC9FwBqk3ItL 29-736-951-3710 unusual, pending theodolites integrate furiously slyly even pinto beans. unusual sheaves sleep befor
8841.59 Supplier#000005750 ROMANIA

100729.00 Manufacturer#5
Erx3lAgu0g62iaHF9x50uMH4EgeN9hEG
29-344-502-5481
excuses after the blithely regular packages mold carefully deposits. regular a
8781.71 Supplier#000003121
ROMANIA
13120.00 Manufacturer#5
wNqTogx238ZYCamFb,50v,bj
4IbNFW9Bvw1xP 29-707-291-5144
packages are quickly after the final, even packages. furiously regular
8754.24 Supplier#000009407
UNITED KINGDOM
179406.00 Manufacturer#4
CHRCbkaWcf5B
33-903-970-9604
regular dependencies haggle across the carefully bold
8691.06 Supplier#000004429
UNITED KINGDOM
126892.00 Manufacturer#2
k,BQms5UhoAF1B2Asi,fLib
33-964-337-5038
quickly special foxes against the furiously silent platelets wake quickly after t
8655.99 Supplier#000006330
RUSSIA
193810.00 Manufacturer#2
UozlaENr0ytKe2w6CeIEWFWn iO3S8Rae7Ou
32-561-198-3705
blithely even packages alongside
8638.36 Supplier#000002920
RUSSIA
75398.00 Manufacturer#1
Je2a8bszf3L
32-122-621-7549
express deposits wake. furiously silent requests wake carefully silent instru
8638.36 Supplier#000002920
RUSSIA
170402.00 Manufacturer#3
Je2a8bszf3L
32-122-621-7549
express deposits wake. furiously silent requests wake carefully silent instru
8607.69 Supplier#000006003
UNITED KINGDOM
76002.00 Manufacturer#2
EH9wADcEiuenM0NR08zDwMidw,52Y2RyIL
EiA 33-416-807-5206
always special foxes wake slyly bold, ironic accounts. ironic instructions affix carefull
8569.52 Supplier#000005936
RUSSIA
5935.00 Manufacturer#5
jXaNZ6vwnEWJ2ksLZJpjtgt0bY2a3AU
32-644-251-7916
packages sleep furiously. special requests about the fluffily even accounts detect
8564.12 Supplier#000000033
GERMANY
110032.00 Manufacturer#1
gfeKpYw3400L0SDywXA6Ya1Qmq1w6YB9f3
R 17-138-897-9374
ironic instructions are. special pearls above
8553.82 Supplier#000003979
ROMANIA
143978.00 Manufacturer#4
BfmVhCAnCMY3jzpjUMy4CNWs9
HzpdQR7INJU 29-124-646-4897
express, ironic pinto beans cajole around the express, even packages. qu
8517.23 Supplier#000009529
RUSSIA
37025.00 Manufacturer#5
e44R8o7JAIS9iMcr
32-565-297-8775
furiously silent requests cajole furiously furiously ironic foxes. slyly express p
8517.23 Supplier#000009529
RUSSIA
59528.00 Manufacturer#2
e44R8o7JAIS9iMcr
32-565-297-8775
furiously silent requests cajole furiously furiously ironic foxes. slyly express p
8503.70 Supplier#000006830

RUSSIA
44325.00 Manufacturer#4
BC4WFCYRUZyaIgchU 4S
32-147-878-5069
quickly regular excuses detect evenly around
8457.09 Supplier#000009456
UNITED KINGDOM
19455.00 Manufacturer#1
7SBhZs8gP1cJjT0Qf433YBk
33-858-440-4349
carefully final accounts sleep blithely special foxes. slyly regular pinto beans alon
8441.40 Supplier#000003817
FRANCE
141302.00 Manufacturer#2
hU3fz3xL78
16-339-356-5115
blithely blithe ideas are
8432.89 Supplier#000003990
RUSSIA
191470.00 Manufacturer#1
wehBBp1RQbfxAYDASS75MsywmsKHRVdkr
vNe6m 32-839-509-9301
final requests along the blithely ironic packages kindle against the carefully fina
8431.40 Supplier#000002675
ROMANIA
5174.00 Manufacturer#1
HJFStOu9R5NGPOegKhgbzBdyvrG2yh8w
29-474-643-1443
express, final deposits cajole carefully. stealthily unusual requests
8407.04 Supplier#000005406
RUSSIA
162889.00 Manufacturer#4
j7 gYF5RW8DC5UrjKC
32-626-152-4621
quickly final sheaves boost. car
8386.08 Supplier#000008518
FRANCE
36014.00 Manufacturer#3
2jqzqqAVe9crMVGP,n9nTsQXulNLTUYoJjEDc
qWV 16-618-780-7481
slyly ironic theodolites are slyly. dogged, pendin
8376.52 Supplier#000005306
UNITED KINGDOM
190267.00 Manufacturer#5
9t8Y8
QqSIsoADPt6NLdk,TP5zyRx41oBUlgoGc9
33-632-514-7931
furiously even instructions integrate during the furiously regular re
8348.74 Supplier#000008851
FRANCE
66344.00 Manufacturer#4
nWxi7GwEbjhw1
16-796-240-2472
ironic instructions nag slyly against the slyly even theodolites. requests alongside of
8338.58 Supplier#000007269
FRANCE
17268.00 Manufacturer#4
ZwhJSwABUoiB04,3
16-267-277-4365
ruthlessly regular asymptotes a
8328.46 Supplier#000001744
ROMANIA
69237.00 Manufacturer#5
oLo3fV64q2,FKHa3p,qHnS7Yzv,ps8
29-330-728-5873
blithely silent excuses are slyly above the furiously even courts
8307.93 Supplier#000003142
GERMANY
18139.00 Manufacturer#1
dqblvV8dCNAorGlJ
17-595-447-6026
theodolites sleep blithely carefully regular warhorses. slyly regular ins
8231.61 Supplier#000009558
RUSSIA
192000.00 Manufacturer#2
mcdgen,yT1iJDHDS5fV
32-762-137-5858
slyly regular theodolites sleep fluffily express depos
8152.61 Supplier#000002731
ROMANIA

15227.00 Manufacturer#4
nluXJCuY1tu
29-805-463-2030
gifts use. slyly silent ideas are carefully beneath the silent instructions. slyly sil
8109.09 Supplier#000009186
FRANCE
99185.00 Manufacturer#1
wgfosrVPexl9pEXWywaqlBMDYYf
16-668-570-1402
quickly pending requests are blithely along the ironic, final requests; instr
8102.62 Supplier#000003347
UNITED KINGDOM
18344.00 Manufacturer#5
m CtXS2S16i
33-454-274-8532
packages grow special orbits. regular theodolites about the carefully pe
8046.07 Supplier#000008780
FRANCE
191222.00 Manufacturer#3
AczzuE0UK9osj ,Lx0Jmh
16-473-215-6395
regular epitaphs integrate slyly.
8042.09 Supplier#000003245
RUSSIA
135705.00 Manufacturer#4
Dh8Ikg39onrbOL4DyTfGw8a9oKUX3d9Y
32-836-132-8872
carefully regular instructions integrate blithely silent foxes. furiously express instructions haggl
8042.09 Supplier#000003245
RUSSIA
150729.00 Manufacturer#1
Dh8Ikg39onrbOL4DyTfGw8a9oKUX3d9Y
32-836-132-8872
carefully regular instructions integrate blithely silent foxes. furiously express instructions haggl
7992.40 Supplier#000006108
FRANCE
118574.00 Manufacturer#1
8tBydnTDwUqfBfFV4l3
16-974-998-8937
regular pinto beans are after
7980.65 Supplier#000001288
FRANCE
13784.00 Manufacturer#4
zE,7HgVPrCn
16-646-464-8247
unusual pinto beans cajole furiously according t
7950.37 Supplier#000008101
GERMANY
33094.00 Manufacturer#5
kkYvL6IuvojJgTNG IKkaXQDYgx8ILohj
17-627-663-8014
quickly regular requests are furiously. pending deposits wake
7937.93 Supplier#000009012
ROMANIA
83995.00 Manufacturer#2
iUiTziH,Ek3i4lwSgunXMgrcTzwdb
29-250-925-9690
blithely bold ideas haggle quickly final, regular request
7914.45 Supplier#000001013
RUSSIA
125988.00 Manufacturer#2
riRcntps4KEDtYScjpMIWeYF6mNnR
32-194-698-3365
final, ironic theodolites alongside of the ironic
7912.91 Supplier#000004211
GERMANY
159180.00 Manufacturer#5
2wQRVovHrm3,v03IKzfTd,1PYsFXQFFOG
17-266-947-7315
final requests integrate slyly above the silent, even
7912.91 Supplier#000004211
GERMANY
184210.00 Manufacturer#4
2wQRVovHrm3,v03IKzfTd,1PYsFXQFFOG
17-266-947-7315
final requests integrate slyly above the silent, even
7894.56 Supplier#000007981
GERMANY
85472.00 Manufacturer#4

NSJ96vMROAbeXP
17-963-404-3760
regular, even theodolites integrate carefully. bold,
special theodolites are slyly fluffily iron
7887.08 Supplier#000009792
GERMANY
164759.00 Manufacturer#3
Y28ITVeYriT3kIGdV2K8fSZ V2UqT5H1Otz
17-988-938-4296
pending, ironic packages sleep among the
carefully ironic accounts. quickly final accounts
7871.50 Supplier#000007206
RUSSIA
104695.00 Manufacturer#1
3w fNCnrVmvJjE95sgWZzvW
32-432-452-7731
furiously dogged pinto beans cajole. bold,
express notornis until the slyly pending
7852.45 Supplier#000005864
RUSSIA
8363.00 Manufacturer#4
WCNfBPZeSXh3h,c
32-454-883-3821
blithely regular deposits
7850.66 Supplier#000001518
UNITED KINGDOM
86501.00 Manufacturer#1
ONda3YJiHKJOC
33-730-383-3892
furiously final accounts wake carefully idle
requests. even dolphins wake acc
7843.52 Supplier#000006683
FRANCE
11680.00 Manufacturer#4
2Z0JGkiv01Y00oCFwUGfviIbhzCdy
16-464-517-8943
carefully bold accounts doub

100 rows processed.
Query Processed in 7.82 seconds.

Ended Executing this Stream at Fri Oct 24 05:54:12 2003

Stream Started at 1066946044.67
Stream Ended at 1066946052.49
Stream Processed in 7.82 seconds

SQL statements processed: 1

D.3 3.log

Begin Execution at Fri Oct 24 05:54:12 2003

-- using default substitutions

```
select * from (
select
l_orderkey,
sum(l_extendedprice * (1 - l_discount)) as revenue,
o_orderdate,
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( '1995-03-15', 'YYYY-MM-DD')
and l_shipdate > to_date( '1995-03-15', 'YYYY-MM-DD')
group by
l_orderkey,
```

```
o_orderdate,
o_shippriority
order by
revenue desc,
o_orderdate)
where rownum <= 10

L_ORDERKEY                    REVENUE
O_ORDERDATE  O_SHIPPRIORITY
2456423.00                    406181.01
1995-03-05    0.00
3459808.00                    405838.70
1995-03-04    0.00
492164.00                     390324.06
1995-02-19    0.00
1188320.00                    384537.94
1995-03-09    0.00
2435712.00                    378673.06
1995-02-26    0.00
4878020.00                    378376.80
1995-03-12    0.00
5521732.00                    375153.92
1995-03-13    0.00
2628192.00                    373133.31
1995-02-22    0.00
993600.00                     371407.46
1995-03-05    0.00
2300070.00                    367371.15
1995-03-13    0.00


10 rows processed.
Query Processed in 10.11 seconds.


Ended Executing this Stream at Fri Oct 24
05:54:22 2003


Stream Started at 1066946052.53
Stream Ended at 1066946062.64
Stream Processed in 10.11 seconds
```

```
SQL statements processed: 1
```

D.4 4.log

```
Begin Execution at Fri Oct 24 05:54:22 2003


-- using default substitutions

select
o_orderpriority,
count(*) as order_count
from
orders
where
o_orderdate >= to_date( '1993-07-01',
'YYYY-MM-DD')
and o_orderdate <
add_months(to_date( '1993-07-01',
'YYYY-MM-DD'),3)
and exists (
select
*
from
lineitem
where
l_orderkey = o_orderkey
and l_commitdate < l_receiptdate
)
group by
o_orderpriority
order by
o_orderpriority

O_ORDERPRIORITY  ORDER_COUNT
1-URGENT          10594.00
2-HIGH            10476.00
3-MEDIUM          10410.00
```

```
4-NOT SPECIFIED 10556.00
5-LOW           10487.00


5 rows processed.
Query Processed in 8.88 seconds.


Ended Executing this Stream at Fri Oct 24 05:54:31 2003


Stream Started at 1066946062.68
Stream Ended at 1066946071.56
Stream Processed in 8.88 seconds


SQL statements processed: 1
```

D.5 5.log

```
Begin Execution at Fri Oct 24 05:54:31 2003


-- using default substitutions


select
n_name,
sum(l_extendedprice * (1 - l_discount)) as
revenue
from
customer,
orders,
lineitem,
supplier,
nation,
region
where
c_custkey = o_custkey
and l_orderkey = o_orderkey
and l_suppkey = s_suppkey
and c_nationkey = s_nationkey
and s_nationkey = n_nationkey
and n_regionkey = r_regionkey
and r_name = 'ASIA'
and o_orderdate >= to_date( '1994-01-01',
'YYYY-MM-DD')
and o_orderdate <
add_months(to_date( '1994-01-01',
'YYYY-MM-DD'), 12)
group by
n_name
order by
revenue desc


N_NAME                    REVENUE
INDONESIA                 55502041.17
VIETNAM                   55295087.00
CHINA                     53724494.26
INDIA                     52035512.00
JAPAN                     45410175.70


5 rows processed.
Query Processed in 10.71 seconds.


Ended Executing this Stream at Fri Oct 24 05:54:42 2003


Stream Started at 1066946071.61
Stream Ended at 1066946082.32
Stream Processed in 10.71 seconds


SQL statements processed: 1
```

D.6 6.log

```
Begin Execution at Fri Oct 24 05:54:42 2003


-- using default substitutions


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 .06 - 0.01 and .06 + 0.01
and l_quantity < 24

REVENUE
123141078.23



1 row processed.
Query Processed in 1.29 seconds.


Ended Executing this Stream at Fri Oct 24
05:54:43 2003


Stream Started at 1066946082.35
Stream Ended at 1066946083.65
Stream Processed in 1.29 seconds


SQL statements processed: 1
```

D.7 7.log

```
Begin Execution at Fri Oct 24 05:54:43 2003


-- using default substitutions


select
supp_nation,
cust_nation,
l_year,
sum(volume) as revenue
from
(
select
n1.n_name as supp_nation,
n2.n_name as cust_nation,
to_number (to_char
(l_shipdate,'yyyy')) as l_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 = 'FRANCE' and n2.n_name =
'GERMANY')
or (n1.n_name = 'GERMANY' and n2.n_name =
'FRANCE')
)
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,
l_year
order by
supp_nation,
cust_nation,
l_year

SUPP_NATION
CUST_NATION                    L_YEAR
REVENUE
FRANCE                         GERMANY
1995.00
54639732.73
FRANCE                         GERMANY
1996.00
54633083.31
GERMANY                        FRANCE
1995.00
52531746.67
GERMANY                        FRANCE
1996.00
52520549.02


4 rows processed.
Query Processed in 11.10 seconds.


Ended Executing this Stream at Fri Oct 24
05:54:54 2003


Stream Started at 1066946083.69
Stream Ended at 1066946094.78
Stream Processed in 11.10 seconds


SQL statements processed: 1
```

D.8 8.log

```
Begin Execution at Fri Oct 24 05:54:54 2003


-- using default substitutions

select
o_year,
sum(case when nation='BRAZIL' then volume
else 0 end )/ sum(volume)
as mkt_share
from
(
select
to_number (to_char (o_orderdate, 'yyyy')) as
o_year,
l_extendedprice * (1 - l_discount) as volume,
n2.n_name as nation
from
part,
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 = 'AMERICA'
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 = 'ECONOMY ANODIZED STEEL'
) all_nations
group by
o_year
order by
o_year

O_YEAR                  MKT_SHARE
1995.00                 0.03
1996.00                 0.04


2 rows processed.
Query Processed in 14.91 seconds.


Ended Executing this Stream at Fri Oct 24 05:55:09 2003


Stream Started at 1066946094.82
Stream Ended at 1066946109.73
Stream Processed in 14.91 seconds


SQL statements processed: 1
```

D.9 9.log

```
Begin Execution at Fri Oct 24 05:55:09 2003


-- using default substitutions


select
nation,
o_year,
sum(amount) as sum_profit
from
(
select
n_name as nation,
to_number (to_char (o_orderdate, 'yyyy')) as o_year,
l_extendedprice * (1 - l_discount) - ps_supplycost * l_quantity as amount
from
part,
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,
o_year
order by
nation,
o_year desc


NATION                  O_YEAR
SUM_PROFIT
ALGERIA                 1998.00
31342867.23
ALGERIA                 1997.00
57138193.02
ALGERIA                 1996.00
56140140.13
ALGERIA                 1995.00
53051469.65
ALGERIA                 1994.00
```

53867582.13		52644504.07	
ALGERIA	1993.00	CANADA	1992.00
54942718.13		53932871.70	
ALGERIA	1992.00	CHINA	1998.00
54628034.71		31075466.16	
ARGENTINA	1998.00	CHINA	1997.00
30211185.71		50551874.45	
ARGENTINA	1997.00	CHINA	1996.00
50805741.75		51039293.88	
ARGENTINA	1996.00	CHINA	1995.00
51923746.58		49287534.62	
ARGENTINA	1995.00	CHINA	1994.00
49298625.77		50851090.07	
ARGENTINA	1994.00	CHINA	1993.00
50835610.11		54229629.83	
ARGENTINA	1993.00	CHINA	1992.00
51646079.18		52400529.37	
ARGENTINA	1992.00	EGYPT	1998.00
50410314.99		29054433.39	
BRAZIL	1998.00	EGYPT	1997.00
27217924.38		50627611.45	
BRAZIL	1997.00	EGYPT	1996.00
48378669.20		49542212.84	
BRAZIL	1996.00	EGYPT	1995.00
50482870.36		48311550.32	
BRAZIL	1995.00	EGYPT	1994.00
47623383.63		49790644.74	
BRAZIL	1994.00	EGYPT	1993.00
47840165.73		48904292.97	
BRAZIL	1993.00	EGYPT	1992.00
49054694.04		49434932.62	
BRAZIL	1992.00	ETHIOPIA	1998.00
48667639.08		28040717.27	
CANADA	1998.00	ETHIOPIA	1997.00
30379833.77		47455009.87	
CANADA	1997.00	ETHIOPIA	1996.00
50465052.31		46491097.57	
CANADA	1996.00	ETHIOPIA	1995.00
52560501.39		46804449.30	
CANADA	1995.00	ETHIOPIA	1994.00
52375332.81		48516143.92	
CANADA	1994.00	ETHIOPIA	1993.00
52600364.66		46551891.56	
CANADA	1993.00	ETHIOPIA	1992.00

44934648.64		27672340.00	
FRANCE	1998.00	INDONESIA	1997.00
32226407.84		50512145.73	
FRANCE	1997.00	INDONESIA	1996.00
47121485.86		51653060.12	
FRANCE	1996.00	INDONESIA	1995.00
47263135.50		51508779.59	
FRANCE	1995.00	INDONESIA	1994.00
47275997.57		52817950.32	
FRANCE	1994.00	INDONESIA	1993.00
47067209.33		47959994.96	
FRANCE	1993.00	INDONESIA	1992.00
51163370.11		51776605.03	
FRANCE	1992.00	IRAN	1998.00
47846235.33		29065736.24	
GERMANY	1998.00	IRAN	1997.00
28624942.66		50042063.05	
GERMANY	1997.00	IRAN	1996.00
49309074.88		50926653.19	
GERMANY	1996.00	IRAN	1995.00
49918683.17		51249667.65	
GERMANY	1995.00	IRAN	1994.00
52650718.72		50337085.87	
GERMANY	1994.00	IRAN	1993.00
50346900.42		51730763.49	
GERMANY	1993.00	IRAN	1992.00
50991895.81		49955856.56	
GERMANY	1992.00	IRAQ	1998.00
48274126.10		31624551.00	
INDIA	1998.00	IRAQ	1997.00
29943144.35		55121749.02	
INDIA	1997.00	IRAQ	1996.00
50665453.23		55897663.79	
INDIA	1996.00	IRAQ	1995.00
50283092.29		54815472.52	
INDIA	1995.00	IRAQ	1994.00
50006774.64		54408516.13	
INDIA	1994.00	IRAQ	1993.00
48995190.76		53633167.98	
INDIA	1993.00	IRAQ	1992.00
50286902.85		55891939.34	
INDIA	1992.00	JAPAN	1998.00
50850329.40		27934179.67	
INDONESIA	1998.00	JAPAN	1997.00

44517162.55		45558221.75	
JAPAN	1996.00	MOROCCO	1995.00
42545606.12		47851318.89	
JAPAN	1995.00	MOROCCO	1994.00
43749356.40		46272172.94	
JAPAN	1994.00	MOROCCO	1993.00
44840243.07		46764326.18	
JAPAN	1993.00	MOROCCO	1992.00
44660015.53		48122783.58	
JAPAN	1992.00	MOZAMBIQUE	1998.00
45410249.12		30712392.01	
JORDAN	1998.00	MOZAMBIQUE	1997.00
26901488.58		50316528.76	
JORDAN	1997.00	MOZAMBIQUE	1996.00
45471878.41		51640320.25	
JORDAN	1996.00	MOZAMBIQUE	1995.00
46794325.79		50693774.51	
JORDAN	1995.00	MOZAMBIQUE	1994.00
45178828.58		49253277.63	
JORDAN	1994.00	MOZAMBIQUE	1993.00
45333636.51		49153016.54	
JORDAN	1993.00	MOZAMBIQUE	1992.00
47971496.10		48247551.85	
JORDAN	1992.00	PERU	1998.00
44717239.18		29326102.32	
KENYA	1998.00	PERU	1997.00
28597614.34		49753780.40	
KENYA	1997.00	PERU	1996.00
47949733.73		50935170.29	
KENYA	1996.00	PERU	1995.00
46886924.62		53309883.41	
KENYA	1995.00	PERU	1994.00
46072338.76		50643531.80	
KENYA	1994.00	PERU	1993.00
45772061.17		51584622.00	
KENYA	1993.00	PERU	1992.00
46308728.23		47523899.05	
KENYA	1992.00	ROMANIA	1998.00
47257780.84		30368667.40	
MOROCCO	1998.00	ROMANIA	1997.00
26732115.58		50365683.85	
MOROCCO	1997.00	ROMANIA	1996.00
45637304.25		49598999.01	
MOROCCO	1996.00	ROMANIA	1995.00

47537642.87
ROMANIA 1994.00
51455283.01
ROMANIA 1993.00
50407136.89
ROMANIA 1992.00
48185385.13
RUSSIA 1998.00
28322384.03
RUSSIA 1997.00
50106685.18
RUSSIA 1996.00
51753342.43
RUSSIA 1995.00
49215820.36
RUSSIA 1994.00
52205666.44
RUSSIA 1993.00
51860230.03
RUSSIA 1992.00
53251677.15
SAUDI ARABIA 1998.00
31541259.81
SAUDI ARABIA 1997.00
52438750.81
SAUDI ARABIA 1996.00
52543737.82
SAUDI ARABIA 1995.00
52938696.53
SAUDI ARABIA 1994.00
51389601.97
SAUDI ARABIA 1993.00
52937508.88
SAUDI ARABIA 1992.00
54843459.64
UNITED KINGDOM 1998.00
28494874.00
UNITED KINGDOM 1997.00
49381810.90
UNITED KINGDOM 1996.00
51386853.96
UNITED KINGDOM 1995.00
51509586.79
UNITED KINGDOM 1994.00
48086499.71
UNITED KINGDOM 1993.00
49166827.22
UNITED KINGDOM 1992.00
49349122.08
UNITED STATES 1998.00
25126238.95
UNITED STATES 1997.00
50077306.42
UNITED STATES 1996.00
48048649.47
UNITED STATES 1995.00
48809032.42
UNITED STATES 1994.00
49296747.18
UNITED STATES 1993.00
48029946.80
UNITED STATES 1992.00
48671944.50
VIETNAM 1998.00
30442736.06
VIETNAM 1997.00
50309179.79
VIETNAM 1996.00
50488161.41
VIETNAM 1995.00
49658284.61
VIETNAM 1994.00
50596057.26
VIETNAM 1993.00
50953919.15
VIETNAM 1992.00
49613838.32

175 rows processed.
Query Processed in 13.64 seconds.

Ended Executing this Stream at Fri Oct 24 05:55:23 2003

Stream Started at 1066946109.77

```
Stream Ended at 1066946123.41
Stream Processed in 13.64 seconds


SQL statements processed: 1
```

D.10 10.log

```
Begin Execution at Fri Oct 24 05:55:23 2003


-- using default substitutions


select * from (
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)
where rownum <= 20

C_CUSTKEY C_NAME
REVENUE
C_ACCTBAL N_NAME
C_ADDRESS
C_PHONE
C_COMMENT
57040.00
Customer#000057040 734235.25
632.87 JAPAN
Eioyzjf4pp
22-895-641-3466
requests sleep blithely about the furiously i
143347.00
Customer#000143347 721002.69
2557.47 EGYPT
1aReFYv,Kw4
14-742-935-3718
fluffily bold excuses haggle finally after the u
60838.00
Customer#000060838 679127.31
2454.77 BRAZIL
64EaJ5vMAHWJlBOxJklpNc2RJiWE
12-913-494-9813
furiously even pinto beans integrate under the
ruthless foxes; ironic, even dolphins across the
slyl
101998.00
Customer#000101998 637029.57
3790.89 UNITED
KINGDOM
01c9CILnNtfOQYmZj
```

33-593-865-6378
accounts doze blithely! enticing, final deposits sleep blithely special accounts. slyly express accounts pla
125341.00
Customer#000125341 633508.09 4983.51 GERMANY S29ODD6bceU8QSuuEJznkNaK 17-582-695-5962
quickly express requests wake quickly blithely
25501.00
Customer#000025501 620269.78 7725.04 ETHIOPIA W556MXuoiaYCCZamJI,Rn0B4ACUGdkQ8DZ 15-874-808-6793
quickly special requests sleep evenly among the special deposits. special deposi
115831.00
Customer#000115831 596423.87 5098.10 FRANCE rFeBbEEyk dl ne7zV5fDrmiq1oK09wV7pxqCgIc 16-715-386-3788
carefully bold excuses sleep alongside of the thinly idle
84223.00
Customer#000084223 594998.02 528.65 UNITED KINGDOM nAVZCs6BaWap rrM27N 2qBnzc5WBauxbA 33-442-824-8191
pending, final ideas haggle final requests. unusual, regular asymptotes affix according to the even foxes.
54289.00
Customer#000054289 585603.39 5583.02 IRAN vXCxoCsU0Bad5JQI ,oobkZ 20-834-292-4707
express requests sublate blithely regular requests. regular, even ideas solve.
39922.00
Customer#000039922 584878.11 7321.11 GERMANY Zgy4s50l2GKN4pLDPBU8m342gIw6R 17-147-757-8036
even pinto beans haggle. slyly bold accounts inte
6226.00
Customer#000006226 576783.76 2230.09 UNITED KINGDOM 8gPu8,NPGkfyQQ0hcIYUGPIBWc,ybP5g, 33-657-701-3391
quickly final requests against the regular instructions wake blithely final instructions. pa
922.00
Customer#000000922 576767.53 3869.25 GERMANY Az9RFaut7NkPnc5zSD2PwHgVwr4jRzq 17-945-916-9648
boldly final requests cajole blith
147946.00
Customer#000147946 576455.13 2030.13 ALGERIA iANyZHjqhyy7Ajah0pTrYyhJ 10-886-956-3143
furiously even accounts are blithely above the furiousl
115640.00
Customer#000115640 569341.19 6436.10 ARGENTINA Vtgfia9qI 7EpHgecU1X 11-411-543-4901
final instructions are slyly according to the
73606.00
Customer#000073606 568656.86 1785.67 JAPAN xuR0Tro5yChDfOCrjkd2ol 22-437-653-6966
furiously bold orbits about the furiously busy requests wake across the furiously quiet theodolites. d
110246.00
Customer#000110246 566842.98 7763.35 VIETNAM 7KzflgX MDOq7sOkI 31-943-426-9837

```
dolphins sleep blithely among the slyly final
142549.00
Customer#000142549          563537.24
5085.99                     INDONESIA
ChqEoK43OysjdHbtKCp6dKqjNyvvi9
19-955-562-2398
regular, unusual dependencies boost slyly; ironic
attainments nag fluffily into the unusual
packages?
146149.00
Customer#000146149          557254.99
1791.55                     ROMANIA
s87fvzFQpU
29-744-164-6487
silent, unusual requests detect quickly slyly regul
52528.00
Customer#000052528          556397.35
551.79                      ARGENTINA
NFztyTOR10UOJ
11-208-192-3205
unusual requests detect. slyly dogged theodolites
use slyly.  deposit
23431.00
Customer#000023431          554269.54
3381.86                     ROMANIA
HgiV0phqhaIa9aydNoIlb
29-915-458-2654
instructions nag quickly. furiously bold accounts
cajol



20 rows processed.
Query Processed in 10.52 seconds.



Ended Executing this Stream at Fri Oct 24
05:55:33 2003



Stream Started at 1066946123.45
Stream Ended at 1066946133.97
Stream Processed in 10.52 seconds
```

```
SQL statements processed: 1
```

D.11 11.log

```
Begin Execution at Fri Oct 24 05:55:34 2003



-- using default substitutions


select
ps_partkey,
sum(ps_supplycost * ps_availqty) as value
from
partsupp,
supplier,
nation
where
ps_suppkey = s_suppkey
and s_nationkey = n_nationkey
and n_name = 'GERMANY'
group by
ps_partkey having
sum(ps_supplycost * ps_availqty) > (
select
sum(ps_supplycost * ps_availqty) *
0.0001000000
from
partsupp,
supplier,
nation
where
ps_suppkey = s_suppkey
and s_nationkey = n_nationkey
and n_name = 'GERMANY'
)
order by
value desc
```

PS_PARTKEY	VALUE	9035.00	12863828.70
129760.00	17538456.86	144616.00	12853549.30
166726.00	16503353.92	176723.00	12832309.74
191287.00	16474801.97	170884.00	12792136.58
161758.00	16101755.54	29790.00	12723300.33
34452.00	15983844.72	95213.00	12555483.73
139035.00	15907078.34	183873.00	12550533.05
9403.00	15451755.62	171235.00	12476538.30
154358.00	15212937.88	21533.00	12437821.32
38823.00	15064802.86	17290.00	12432159.50
85606.00	15053957.15	156397.00	12260623.50
33354.00	14408297.40	122611.00	12222812.98
154747.00	14407580.68	139155.00	12220319.25
82865.00	14235489.78	146316.00	12215800.61
76094.00	14094247.04	171381.00	12199734.52
222.00	13937777.74	198633.00	12078226.95
121271.00	13908336.00	167417.00	12046637.62
55221.00	13716120.47	59512.00	12043468.76
22819.00	13666434.28	31688.00	12034893.64
76281.00	13646853.68	159586.00	12001505.84
85298.00	13581154.93	8993.00	11963814.30
85158.00	13554904.00	120302.00	11857707.55
139684.00	13535538.72	43536.00	11779340.52
31034.00	13498025.25	9552.00	11776909.16
87305.00	13482847.04	86223.00	11772205.08
10181.00	13445148.75	53776.00	11758669.65
62323.00	13411824.30	131285.00	11616953.74
26489.00	13377256.38	91628.00	11611114.83
96493.00	13339057.83	169644.00	11567959.72
56548.00	13329014.97	182299.00	11567462.05
55576.00	13306843.35	33107.00	11453818.76
159751.00	13306614.48	104184.00	11436657.44
92406.00	13287414.50	67027.00	11419127.14
182636.00	13223726.74	176869.00	11371451.71
199969.00	13135288.21	30885.00	11369674.79
62865.00	13001926.94	54420.00	11345076.88
7284.00	12945298.19	72240.00	11313951.05
197867.00	12944510.52	178708.00	11294635.17
11562.00	12931575.51	81298.00	11273686.13
75165.00	12916918.12	158324.00	11243442.72
97175.00	12911283.50	117095.00	11242535.24
140840.00	12896562.23	176793.00	11237733.38
65241.00	12890600.46	86091.00	11177793.79
166120.00	12876927.22	116033.00	11145434.36

129058.00	11119112.20	195810.00	10413625.20
193714.00	11104706.39	76673.00	10391977.18
117195.00	11077217.96	97305.00	10390890.57
49851.00	11043701.78	134210.00	10387210.02
19791.00	11030662.62	188536.00	10386529.92
75800.00	11012401.62	122255.00	10335760.32
161562.00	10996371.69	2682.00	10312966.10
10119.00	10980015.75	43814.00	10303086.61
39185.00	10970042.56	34767.00	10290405.18
47223.00	10950022.13	165584.00	10273705.89
175594.00	10942923.05	2231.00	10270415.55
111295.00	10893675.61	111259.00	10263256.56
155446.00	10852764.57	195578.00	10239795.82
156391.00	10839810.38	21093.00	10217531.30
40884.00	10837234.19	29856.00	10216932.54
141288.00	10837130.21	133686.00	10213345.76
152388.00	10830977.82	87745.00	10185509.40
33449.00	10830858.72	135153.00	10179379.70
149035.00	10826130.02	11773.00	10167410.84
162620.00	10814275.68	76316.00	10165151.70
118324.00	10791788.10	123076.00	10161225.78
38932.00	10777541.75	91894.00	10130462.19
121294.00	10764225.22	39741.00	10128387.52
48721.00	10762582.49	111753.00	10119780.98
63342.00	10740132.60	142729.00	10104748.89
5614.00	10724668.80	116775.00	10097750.42
62266.00	10711143.10	102589.00	10034784.36
100202.00	10696675.55	186268.00	10012181.57
197741.00	10688560.72	44545.00	10000286.48
169178.00	10648522.80	23307.00	9966577.50
5271.00	10639392.65	124281.00	9930018.90
34499.00	10584177.10	69604.00	9925730.64
71108.00	10569117.56	21971.00	9908982.03
137132.00	10539880.47	58148.00	9895894.40
78451.00	10524873.24	16532.00	9886529.90
150827.00	10503810.48	159180.00	9883744.43
107237.00	10488030.84	74733.00	9877582.88
101727.00	10473558.10	35173.00	9858275.92
58708.00	10466280.44	7116.00	9856881.02
89768.00	10465477.22	124620.00	9838589.14
146493.00	10444291.58	122108.00	9829949.35
55424.00	10444006.48	67200.00	9828690.69
16560.00	10425574.74	164775.00	9821424.44
133114.00	10415097.90	9039.00	9816447.72

14912.00	9803102.20	52594.00	9508325.76
190906.00	9791315.70	60960.00	9498843.06
130398.00	9781674.27	70272.00	9495775.62
119310.00	9776927.21	44050.00	9495515.36
10132.00	9770930.78	152213.00	9494756.96
107211.00	9757586.25	121203.00	9492601.30
113958.00	9757065.50	70114.00	9491012.30
37009.00	9748362.69	167588.00	9484741.11
66746.00	9743528.76	136455.00	9476241.78
134486.00	9731922.00	4357.00	9464355.64
15945.00	9731096.45	6786.00	9463632.57
55307.00	9717745.80	61345.00	9455336.70
56362.00	9714922.83	160826.00	9446754.84
57726.00	9711792.10	71275.00	9440138.40
57256.00	9708621.00	77746.00	9439118.35
112292.00	9701653.08	91289.00	9437472.00
87514.00	9699492.53	56723.00	9435102.16
174206.00	9680562.02	86647.00	9434604.18
72865.00	9679043.34	131234.00	9432120.00
114357.00	9671017.44	198129.00	9427651.36
112807.00	9665019.21	165530.00	9426193.68
115203.00	9661018.73	69233.00	9425053.92
177454.00	9658906.35	6243.00	9423304.66
161275.00	9634313.71	90110.00	9420422.70
61893.00	9617095.44	191980.00	9419368.36
122219.00	9604888.20	38461.00	9419316.07
183427.00	9601362.58	167873.00	9419024.49
59158.00	9599705.96	159373.00	9416950.15
61931.00	9584918.98	128707.00	9413428.50
5532.00	9579964.14	45267.00	9410863.78
20158.00	9576714.38	48460.00	9409793.93
167199.00	9557413.08	197672.00	9406887.68
38869.00	9550279.53	60884.00	9403442.40
86949.00	9541943.70	15209.00	9403245.31
198544.00	9538613.92	138049.00	9401262.10
193762.00	9538238.94	199286.00	9391770.70
108807.00	9536247.16	19629.00	9391236.40
168324.00	9535647.99	134019.00	9390615.15
115588.00	9532195.04	169475.00	9387639.58
141372.00	9529702.14	165918.00	9379510.44
175120.00	9526068.66	135602.00	9374251.54
163851.00	9522808.83	162323.00	9367566.51
160954.00	9520359.45	96277.00	9360850.68
117757.00	9517882.80	98336.00	9359671.29

119781.00	9356395.73	147248.00	9205654.95
34440.00	9355365.00	61372.00	9205228.76
57362.00	9355180.10	52970.00	9204415.95
167236.00	9352973.84	26430.00	9203710.51
38463.00	9347530.94	28504.00	9201669.20
86749.00	9346826.44	25810.00	9198878.50
170007.00	9345699.90	125329.00	9198688.50
193087.00	9343744.00	167867.00	9194022.72
150383.00	9332576.75	134767.00	9191444.72
60932.00	9329582.02	127745.00	9191271.56
128420.00	9328206.35	69208.00	9187110.00
162145.00	9327722.88	155222.00	9186469.16
55686.00	9320304.40	196916.00	9182995.82
163080.00	9304916.96	195590.00	9176353.12
160583.00	9303515.92	169155.00	9175176.09
118153.00	9298606.56	81558.00	9171946.50
152634.00	9282184.57	185136.00	9171293.04
84731.00	9276586.92	114790.00	9168509.10
119989.00	9273814.20	194142.00	9165836.61
114584.00	9269698.65	167639.00	9161165.00
131817.00	9268570.08	11241.00	9160789.46
29068.00	9256583.88	82628.00	9160155.54
44116.00	9255922.00	41399.00	9148338.00
115818.00	9253311.91	30755.00	9146196.84
103388.00	9239218.08	6944.00	9143574.58
186118.00	9236209.12	6326.00	9138803.16
155809.00	9235410.84	101296.00	9135657.62
147003.00	9234847.99	181479.00	9121093.30
27769.00	9232511.64	76898.00	9120983.10
112779.00	9231927.36	64274.00	9118745.25
124851.00	9228982.68	175826.00	9117387.99
158488.00	9227216.40	142215.00	9116876.88
83328.00	9224792.20	103415.00	9113128.62
136797.00	9222927.09	119765.00	9110768.79
141730.00	9216370.68	107624.00	9108837.45
87304.00	9215695.50	84215.00	9105257.36
156004.00	9215557.90	73774.00	9102651.92
140740.00	9215329.20	173972.00	9102069.00
100648.00	9212185.08	69817.00	9095513.88
174774.00	9211718.00	86943.00	9092253.00
37644.00	9211578.60	138859.00	9087719.30
48807.00	9209496.24	162273.00	9085296.48
95940.00	9207948.40	175945.00	9080401.21
141586.00	9206699.22	16836.00	9075715.44

70224.00	9075265.95	177372.00	8954873.64
139765.00	9074755.89	137994.00	8950916.79
30319.00	9073233.10	84019.00	8950039.98
3851.00	9072657.24	40389.00	8946158.20
181271.00	9070631.52	69187.00	8941054.14
162184.00	9068835.78	4863.00	8939044.92
81683.00	9067258.47	50465.00	8930503.14
153028.00	9067010.51	43686.00	8915543.84
123324.00	9061870.95	131352.00	8909053.59
186481.00	9058608.30	198916.00	8906940.03
167680.00	9052908.76	135932.00	8905282.95
165293.00	9050545.70	104673.00	8903682.00
122148.00	9046298.17	152308.00	8903244.08
138604.00	9045840.80	135298.00	8900323.20
78851.00	9044822.60	156873.00	8899429.10
137280.00	9042355.34	157454.00	8897339.20
8823.00	9040855.10	75415.00	8897068.09
163900.00	9040848.48	46325.00	8895569.09
75600.00	9035392.45	1966.00	8895117.06
81676.00	9031999.40	24576.00	8895034.75
46033.00	9031460.58	19425.00	8890156.60
194917.00	9028500.00	169735.00	8890085.56
133936.00	9026949.02	32225.00	8889829.28
33182.00	9024971.10	124537.00	8889770.71
34220.00	9021485.39	146327.00	8887836.23
20118.00	9019942.60	121562.00	8887740.40
178258.00	9019881.66	44731.00	8882444.95
15560.00	9017687.28	93141.00	8881850.88
111425.00	9016198.56	187871.00	8873506.18
95942.00	9015585.12	71709.00	8873057.28
132709.00	9015240.15	151913.00	8869321.17
39731.00	9014746.95	33786.00	8868955.39
154307.00	9012571.20	35902.00	8868126.06
23769.00	9008157.60	23588.00	8867769.90
93328.00	9007211.20	24508.00	8867616.00
142826.00	8998297.44	161282.00	8866661.43
188792.00	8996014.00	188061.00	8862304.00
68703.00	8994982.22	132847.00	8862082.00
145280.00	8990941.05	166843.00	8861200.80
150725.00	8985686.16	30609.00	8860214.73
172046.00	8982469.52	56191.00	8856546.96
70476.00	8967629.50	160740.00	8852685.43
124988.00	8966805.22	71229.00	8846106.99
17937.00	8963319.76	91208.00	8845541.28

10995.00	8845306.56	30235.00	8741611.00
78094.00	8839938.29	26391.00	8741399.64
36489.00	8838538.10	191816.00	8740258.72
198437.00	8836494.84	47616.00	8737229.68
151693.00	8833807.64	152101.00	8734432.76
185367.00	8829791.37	163784.00	8730514.34
65682.00	8820622.89	5134.00	8728424.64
65421.00	8819329.24	155241.00	8725429.86
122225.00	8816821.86	188814.00	8724182.40
85330.00	8811013.16	140782.00	8720378.75
64555.00	8810643.12	153141.00	8719407.51
104188.00	8808211.02	169373.00	8718609.06
54411.00	8805703.40	41335.00	8714773.80
39438.00	8805282.56	197450.00	8714617.32
70795.00	8800060.92	87004.00	8714017.79
20383.00	8799073.28	181804.00	8712257.76
21952.00	8798624.19	122814.00	8711119.14
63584.00	8796590.00	109939.00	8709193.16
158768.00	8796422.95	98094.00	8708780.04
166588.00	8796214.38	74630.00	8708040.75
120600.00	8793558.06	197291.00	8706519.09
157202.00	8788287.88	184173.00	8705467.45
55358.00	8786820.75	192175.00	8705411.12
168322.00	8786670.73	19471.00	8702536.12
25143.00	8786324.80	18052.00	8702155.70
5368.00	8786274.14	135560.00	8698137.72
114025.00	8786201.12	152791.00	8697325.80
97744.00	8785315.94	170953.00	8696909.19
164327.00	8784503.86	116137.00	8696687.17
76542.00	8782613.28	7722.00	8696589.40
4731.00	8772846.70	49788.00	8694846.71
157590.00	8772006.45	13252.00	8694822.42
154276.00	8771733.91	12633.00	8694559.36
28705.00	8771576.64	193438.00	8690426.72
100226.00	8769455.00	17326.00	8689329.16
179195.00	8769185.16	96124.00	8679794.58
184355.00	8768118.05	143802.00	8676626.48
120408.00	8768011.12	30389.00	8675826.60
63145.00	8761991.96	75250.00	8675257.14
53135.00	8753491.80	72613.00	8673524.94
173071.00	8750508.80	123520.00	8672456.25
41087.00	8749436.79	325.00	8667741.28
194830.00	8747438.40	167291.00	8667556.18
43496.00	8743359.30	150119.00	8663403.54

88420.00	8663355.40	5840.00	8570728.74
179784.00	8653021.34	120860.00	8570610.44
130884.00	8651970.00	66692.00	8567540.52
172611.00	8648217.00	135596.00	8563276.31
85373.00	8647796.22	150576.00	8562794.10
122717.00	8646758.54	7500.00	8562393.84
113431.00	8646348.34	107716.00	8561541.56
66015.00	8643349.40	100611.00	8559995.85
33141.00	8643243.18	171192.00	8557390.08
69786.00	8637396.92	107660.00	8556696.60
181857.00	8637393.28	13461.00	8556545.12
122939.00	8636378.00	90310.00	8555131.51
196223.00	8635391.02	141493.00	8553782.93
50532.00	8632648.24	71286.00	8552682.00
58102.00	8632614.54	136423.00	8551300.76
93581.00	8632372.36	54241.00	8550785.25
52804.00	8632109.25	120325.00	8549976.60
755.00	8627091.68	424.00	8547527.10
16597.00	8623357.05	196543.00	8545907.09
119041.00	8622397.00	13042.00	8542717.18
89050.00	8621185.98	58332.00	8536074.69
98696.00	8620784.82	9191.00	8535663.92
94399.00	8620524.00	134357.00	8535429.90
151295.00	8616671.02	96207.00	8534900.60
56417.00	8613450.35	92292.00	8530618.78
121322.00	8612948.23	181093.00	8528303.52
126883.00	8611373.42	105064.00	8527491.60
29155.00	8610163.64	59635.00	8526854.08
114530.00	8608471.74	136974.00	8524351.56
131007.00	8607394.82	126694.00	8522783.37
128715.00	8606833.62	6247.00	8522606.90
72522.00	8601479.98	139447.00	8522521.92
144061.00	8595718.74	96313.00	8520949.92
83503.00	8595034.20	108454.00	8520916.25
112199.00	8590717.44	181254.00	8519496.10
9227.00	8587350.42	71117.00	8519223.00
116318.00	8585910.66	131703.00	8517215.28
41248.00	8585559.64	59312.00	8510568.36
159398.00	8584821.00	2903.00	8509960.35
105966.00	8582308.79	102838.00	8509527.69
137876.00	8580641.30	162806.00	8508906.05
122272.00	8580400.77	41527.00	8508222.36
195717.00	8577278.10	118416.00	8505858.36
165295.00	8571121.92	180203.00	8505024.16

14773.00	8500598.28	163068.00	8431116.40
140446.00	8499514.24	138772.00	8428406.36
199641.00	8497362.59	126821.00	8425180.68
109240.00	8494617.12	22091.00	8420687.88
150268.00	8494188.38	55981.00	8419434.38
45310.00	8492380.65	100960.00	8419403.46
36552.00	8490733.60	172568.00	8417955.21
199690.00	8490145.80	63135.00	8415945.53
185353.00	8488726.68	137651.00	8413170.35
163615.00	8484985.01	191353.00	8413039.84
196520.00	8483545.04	62988.00	8411571.48
133438.00	8483482.35	103417.00	8411541.12
77285.00	8481442.32	12052.00	8411519.28
55824.00	8476893.90	104260.00	8408516.55
76753.00	8475522.12	157129.00	8405730.08
46129.00	8472717.96	77254.00	8405537.22
28358.00	8472515.50	112966.00	8403512.89
9317.00	8472145.32	168114.00	8402764.56
33823.00	8469721.44	49940.00	8402328.20
39055.00	8469145.07	52017.00	8398753.60
91471.00	8468874.56	176179.00	8398087.00
142299.00	8466039.55	100215.00	8395906.61
97672.00	8464119.80	61256.00	8392811.20
134712.00	8461781.79	15366.00	8388907.80
157988.00	8460123.20	109479.00	8388027.20
102284.00	8458652.44	66202.00	8386522.83
73533.00	8458453.32	81707.00	8385761.19
90599.00	8457874.86	51727.00	8385426.40
112160.00	8457863.36	9980.00	8382754.62
124792.00	8457633.70	174403.00	8378575.73
66097.00	8457573.15	54558.00	8378041.92
165271.00	8456969.01	3141.00	8377378.22
146925.00	8454887.91	134829.00	8377105.52
164277.00	8454838.50	145056.00	8376920.76
131290.00	8454811.20	194020.00	8375157.64
179386.00	8450909.90	7117.00	8373982.27
90486.00	8447873.86	120146.00	8373796.20
175924.00	8444421.66	126843.00	8370761.28
185922.00	8442394.88	62117.00	8369493.44
38492.00	8436438.32	111221.00	8367525.81
172511.00	8436287.34	159337.00	8366092.26
139539.00	8434180.29	173903.00	8365428.48
11926.00	8433199.52	136438.00	8364065.45
55889.00	8431449.88	56684.00	8363198.00

137597.00	8363185.94	14545.00	8288395.92
20039.00	8361138.24	75548.00	8288287.20
121326.00	8359635.52	64473.00	8286137.44
48435.00	8352863.10	149553.00	8285714.88
1712.00	8349107.00	151284.00	8283526.65
167190.00	8347238.70	171091.00	8282934.36
32113.00	8346452.04	194256.00	8278985.34
40580.00	8342983.32	952.00	8276136.00
74785.00	8342519.13	121541.00	8275390.26
14799.00	8342236.75	177664.00	8275315.20
177291.00	8341736.83	51117.00	8274504.30
198956.00	8340370.65	66770.00	8273407.80
69179.00	8338465.99	37238.00	8272728.06
118764.00	8337616.56	46679.00	8270486.55
128814.00	8336435.56	165852.00	8268312.60
82729.00	8331766.88	99458.00	8266564.47
152048.00	8330638.99	114519.00	8265493.54
171085.00	8326259.50	7231.00	8264881.50
126730.00	8325974.40	19033.00	8264826.56
77525.00	8323282.50	125123.00	8262732.65
170653.00	8322840.50	18642.00	8261578.99
5257.00	8320350.78	50386.00	8261380.05
67350.00	8318987.56	193770.00	8259578.82
109008.00	8317836.54	7276.00	8258101.60
199043.00	8316603.54	178045.00	8253904.15
139969.00	8316551.54	49033.00	8253696.23
22634.00	8316531.24	187195.00	8251334.58
173309.00	8315750.25	10590.00	8249227.40
10887.00	8315019.36	143779.00	8247057.70
42392.00	8312895.96	35205.00	8245675.17
126040.00	8312623.20	19729.00	8245081.60
101590.00	8304555.42	144946.00	8240479.80
46891.00	8302192.12	123786.00	8239581.24
138721.00	8301745.62	70843.00	8237973.20
113715.00	8301533.20	112437.00	8236907.52
78778.00	8299685.64	5436.00	8236039.57
142908.00	8299447.77	163754.00	8235471.16
64419.00	8297631.80	115945.00	8234811.36
21396.00	8296272.27	27918.00	8233957.88
4180.00	8295646.92	105712.00	8233571.86
63534.00	8295383.67	41007.00	8229431.79
135957.00	8294389.86	40476.00	8226640.41
30126.00	8291920.32	145620.00	8221371.60
158427.00	8288938.00	7771.00	8220413.33

86424.00	8215572.61	51295.00	8156419.20
129137.00	8215478.40	69512.00	8151537.00
76020.00	8210495.36	164274.00	8149869.93
140213.00	8209831.80	130854.00	8145338.85
32379.00	8208338.88	186865.00	8143586.82
130616.00	8207715.75	176629.00	8141411.20
195469.00	8206609.80	193739.00	8141377.77
191805.00	8205147.75	6810.00	8139822.60
90906.00	8200951.20	27732.00	8136724.96
170910.00	8195558.01	50616.00	8134089.82
105399.00	8193122.63	123908.00	8128920.54
123798.00	8192385.97	140994.00	8128470.82
90218.00	8191689.16	99039.00	8128290.78
114766.00	8189339.54	62735.00	8124940.50
11289.00	8187354.72	47829.00	8122796.50
178308.00	8185750.50	192635.00	8122687.57
71271.00	8185519.24	192429.00	8119268.00
1115.00	8184903.38	145812.00	8119165.63
152636.00	8184530.72	42896.00	8118529.80
151619.00	8182909.05	146877.00	8118266.16
116943.00	8181072.69	60882.00	8116095.04
28891.00	8181051.54	18254.00	8114783.04
47049.00	8180955.00	165464.00	8114571.80
158827.00	8180470.90	57936.00	8111927.25
92620.00	8179671.55	52226.00	8110723.32
20814.00	8176953.54	128571.00	8106788.80
179323.00	8176795.55	100308.00	8105837.04
193453.00	8174343.94	8872.00	8102395.62
56888.00	8173342.00	58867.00	8102033.19
28087.00	8169876.30	145153.00	8100222.84
164254.00	8169632.35	172088.00	8098138.20
57661.00	8168848.16	59398.00	8095845.45
7363.00	8167538.05	89395.00	8093576.10
164499.00	8167512.08	171961.00	8093538.00
197557.00	8165940.45	88736.00	8090762.16
5495.00	8164805.22	174053.00	8090350.11
966.00	8163824.79	102237.00	8089103.22
98435.00	8161771.45	43041.00	8086537.90
127227.00	8161344.92	110219.00	8085296.90
194100.00	8160978.78	126738.00	8084199.20
40134.00	8160358.08	44787.00	8083628.40
107341.00	8159952.05	31277.00	8083580.76
6790.00	8158792.66	93595.00	8082188.80
43851.00	8157101.40	189040.00	8080257.21

59851.00	8079024.24	110971.00	8029469.70
175100.00	8077904.01	130395.00	8027463.92
43429.00	8076729.96	7757.00	8026840.37
154199.00	8074940.76	178446.00	8025379.09
60963.00	8073894.40	41295.00	8024785.53
8768.00	8072760.96	100956.00	8024179.30
66095.00	8071421.70	131917.00	8021604.78
111552.00	8068184.48	24224.00	8020463.52
24563.00	8067500.40	2073.00	8020009.64
16167.00	8067495.24	121622.00	8018462.17
12662.00	8067248.85	14357.00	8016906.30
94540.00	8063727.16	135601.00	8016209.44
23308.00	8063463.18	58458.00	8016192.52
27390.00	8062823.25	73036.00	8015799.00
130660.00	8062787.48	184722.00	8015680.31
8608.00	8062411.16	151664.00	8014821.96
181552.00	8062008.30	195090.00	8012680.20
199319.00	8060248.56	162609.00	8011241.00
55475.00	8058850.92	83532.00	8009753.85
142711.00	8057926.58	50166.00	8007137.89
103499.00	8056978.00	181562.00	8006805.96
105943.00	8056698.75	175165.00	8005319.76
8432.00	8053052.16	62500.00	8005316.28
149392.00	8049675.69	36342.00	8004333.40
101248.00	8048855.49	128435.00	8004242.88
140962.00	8047260.70	92516.00	8003836.80
87101.00	8046651.83	30802.00	8003710.88
133107.00	8046476.73	107418.00	8000430.30
45126.00	8045924.40	46620.00	7999778.35
87508.00	8042966.39	191803.00	7994734.15
124711.00	8042722.72	106343.00	7993087.76
173169.00	8042224.41	59362.00	7990397.46
175161.00	8041331.98	8329.00	7990052.90
167787.00	8040075.78	75133.00	7988244.00
3242.00	8038855.53	179023.00	7986829.62
114789.00	8038628.35	135899.00	7985726.64
43833.00	8038545.83	5824.00	7985340.02
141198.00	8035110.72	148579.00	7984889.56
137248.00	8034109.35	95888.00	7984735.72
96673.00	8033491.20	9791.00	7982699.79
32180.00	8032380.72	170437.00	7982370.72
166493.00	8031902.40	39782.00	7977858.24
66959.00	8031839.40	20605.00	7977556.00
85628.00	8029693.44	28682.00	7976960.00

42172.00	7973399.00	196514.00	7927180.70
56137.00	7971405.40	4403.00	7925729.04
64729.00	7970769.72	2267.00	7925649.37
98643.00	7968603.73	45924.00	7925047.68
153787.00	7967535.58	11493.00	7916722.23
8932.00	7967222.19	104478.00	7916253.60
20134.00	7965713.28	166794.00	7913842.00
197635.00	7963507.58	161995.00	7910874.27
80408.00	7963312.17	23538.00	7909752.06
37728.00	7961875.68	41093.00	7909579.92
26624.00	7961772.31	112073.00	7908617.57
44736.00	7961144.10	92814.00	7908262.50
29763.00	7960605.03	88919.00	7907992.50
36147.00	7959463.68	79753.00	7907933.88
146040.00	7957587.66	108765.00	7905338.98
115469.00	7957485.14	146530.00	7905336.60
142276.00	7956790.63	71475.00	7903367.58
181280.00	7954037.35	36289.00	7901946.50
115096.00	7953047.55	61739.00	7900794.00
109650.00	7952258.73	52338.00	7898638.08
93862.00	7951992.24	194299.00	7898421.24
158325.00	7950728.30	105235.00	7897829.94
55952.00	7950387.06	77207.00	7897752.72
122397.00	7947106.27	96712.00	7897575.27
28114.00	7946945.72	10157.00	7897046.25
11966.00	7945197.48	171154.00	7896814.50
47814.00	7944083.00	79373.00	7896186.00
85096.00	7943691.06	113808.00	7893353.88
51657.00	7943593.77	27901.00	7892952.00
196680.00	7943578.89	128820.00	7892882.72
13141.00	7942730.34	25891.00	7890511.20
193327.00	7941036.25	122819.00	7888881.02
152612.00	7940663.71	154731.00	7888301.33
139680.00	7939242.36	101674.00	7879324.60
31134.00	7938318.30	51968.00	7879102.21
45636.00	7937240.85	72073.00	7877736.11
56694.00	7936015.95	5182.00	7874521.73
8114.00	7933921.88		
71518.00	7932261.69		
72922.00	7930400.64	1048 rows processed.	
146699.00	7929167.40	Query Processed in 6.92 seconds.	
92387.00	7928972.67		
186289.00	7928786.19		
95952.00	7927972.78	Ended Executing this Stream at Fri Oct 24	

```
05:55:40 2003


Stream Started at 1066946134.01
Stream Ended at 1066946140.93
Stream Processed in 6.92 seconds


SQL statements processed: 1
```

D.12 12.log

```
Begin Execution at Fri Oct 24 05:55:40 2003


-- using default substitutions


select
        l_shipmode,
        sum(case
                when o_orderpriority = '1-URGENT'
                        or o_orderpriority = '2-HIGH'
                        then 1
                else 0
        end) as high_line_count,
        sum(case
                when o_orderpriority <> '1-URGENT'
                        and o_orderpriority <> '2-HIGH'
                        then 1
                else 0
        end) as low_line_count
from
        orders,
        lineitem
where
        o_orderkey = l_orderkey
        and l_shipmode in ('MAIL', 'SHIP')
        and l_commitdate < l_receiptdate
        and l_shipdate < l_commitdate
        and l_receiptdate >= to_date( '1994-01-01', 'YYYY-MM-DD')
        and l_receiptdate < add_months(to_date ('1994-01-01', 'YYYY-MM-DD'), 12)
group by
        l_shipmode
order by
        l_shipmode

L_SHIPMODE                HIGH_LINE_COUNT
LOW_LINE_COUNT
MAIL                                6202.00
9324.00
SHIP                                6200.00
9262.00


2 rows processed.
Query Processed in 7.39 seconds.


Ended Executing this Stream at Fri Oct 24 05:55:48 2003


Stream Started at 1066946140.98
Stream Ended at 1066946148.37
Stream Processed in 7.39 seconds


SQL statements processed: 1
```

D.13 13.log

```
Begin Execution at Fri Oct 24 05:55:48 2003


-- using default substitutions
```

```
                                                    24.00           2622.00
                                                    25.00           2079.00
select                                              5.00            1972.00
c_count,                                            26.00           1593.00
count(*) as custdist                                27.00           1185.00
from                                                4.00            1033.00
(                                                   28.00           869.00
select                                              29.00           559.00
c_custkey,                                          3.00            398.00
count(o_orderkey) as c_count                        30.00           373.00
from                                                31.00           235.00
customer, orders where                              2.00            144.00
c_custkey = o_custkey(+)                            32.00           128.00
and o_comment(+) not like                           33.00           71.00
'%special%requests%'                                34.00           48.00
group by                                            35.00           33.00
c_custkey                                           1.00            23.00
) c_orders                                          36.00           17.00
group by                                            37.00           7.00
c_count                                             40.00           4.00
order by                                            38.00           4.00
custdist desc,                                      39.00           2.00
c_count desc                                        41.00           1.00

C_COUNT         CUSTDIST
0.00            50004.00        42 rows processed.
9.00            6641.00         Query Processed in 7.72 seconds.
10.00           6566.00
11.00           6058.00
8.00            5949.00         Ended Executing this Stream at Fri Oct 24
12.00           5553.00         05:55:56 2003
13.00           4989.00
19.00           4748.00
7.00            4707.00         Stream Started at 1066946148.41
18.00           4625.00         Stream Ended at 1066946156.13
15.00           4552.00         Stream Processed in 7.72 seconds
17.00           4530.00
14.00           4484.00
20.00           4461.00         SQL statements processed: 1
16.00           4323.00
21.00           4217.00
22.00           3730.00
6.00            3334.00
23.00           3129.00
```

D.14 14.log

```
-- using default substitutions


select
	100.00 * sum(case
		when p_type like 'PROMO%'
			then l_extendedprice * (1 -
l_discount)
		else 0
	end) / sum(l_extendedprice * (1 -
l_discount)) as promo_revenue
from
	lineitem,
	part
where
	l_partkey = p_partkey
	and l_shipdate >= date '1995-09-01'
	and l_shipdate < date '1995-09-01' +
interval '1' month

PROMO_REVENUE
16.38


1 row processed.
Query Processed in 1.75 seconds.


Ended Executing this Stream at Fri Oct 24
05:55:57 2003


Stream Started at 1066946156.17
Stream Ended at 1066946157.91
Stream Processed in 1.75 seconds
```

SQL statements processed: 1

Begin Execution at Fri Oct 24 05:55:56 2003

D.15 15.log

Begin Execution at Fri Oct 24 05:55:57 2003

```
-- using default substitutions


create view revenue0 (supplier_no,
total_revenue) as
select
l_suppkey,
sum(l_extendedprice * (1 - l_discount))
from
lineitem
where
l_shipdate >= to_date( '1996-01-01',
'YYYY-MM-DD')
and l_shipdate < add_months( to_date
('1996-01-01', 'YYYY-MM-DD'), 3)
group by
l_suppkey
Query Processed in 0.01 seconds.


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
8449.00                  Supplier#000008449
Wp34zim9qYFbVctdW
20-469-856-8873  1772627.21


1 row   processed.
Query Processed in 10.49 seconds.



drop view revenue0
Query Processed in 0.03 seconds.


Ended Executing this Stream at Fri Oct 24
05:56:08 2003


Stream Started at 1066946157.95
Stream Ended at 1066946168.49
Stream Processed in 10.53 seconds


SQL statements processed: 3
```

D.16 16.log

```
Begin Execution at Fri Oct 24 05:56:08 2003


-- using default substitutions


select
p_brand,
p_type,
p_size,
count(distinct ps_suppkey) as supplier_cnt
from
partsupp,
part
where
p_partkey = ps_partkey
and p_brand <> 'Brand#45'
and p_type not like 'MEDIUM POLISHED%'
and p_size in (49, 14, 23, 45, 19, 3, 36, 9)
and ps_suppkey not in (
select
s_suppkey
from
supplier
where
s_comment like '%Customer%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#41         MEDIUM BRUSHED TIN
3.00                  28.00
Brand#54         STANDARD BRUSHED
```

```
COPPER 14.00 27.00
Brand#11 STANDARD BRUSHED TIN
23.00 24.00
Brand#11 STANDARD BURNISHED
BRASS 36.00 24.00
Brand#15 MEDIUM ANODIZED NICKEL
3.00 24.00
Brand#15 SMALL ANODIZED BRASS
45.00 24.00
Brand#15 SMALL BURNISHED NICKEL
19.00 24.00
Brand#21 MEDIUM ANODIZED COPPER
3.00 24.00
Brand#22 SMALL BRUSHED NICKEL
3.00 24.00
Brand#22 SMALL BURNISHED BRASS
19.00 24.00
Brand#25 MEDIUM BURNISHED COPPER
36.00 24.00
Brand#31 PROMO POLISHED COPPER
36.00 24.00
Brand#33 LARGE POLISHED TIN
23.00 24.00
Brand#33 PROMO POLISHED STEEL
14.00 24.00
Brand#35 PROMO BRUSHED NICKEL
14.00 24.00
Brand#41 ECONOMY BRUSHED STEEL
9.00 24.00
Brand#41 ECONOMY POLISHED TIN
19.00 24.00
Brand#41 LARGE PLATED COPPER
36.00 24.00
Brand#42 ECONOMY PLATED BRASS
3.00 24.00
Brand#42 STANDARD POLISHED TIN
49.00 24.00

---------- <rows truncated>
----------

18314 rows processed.
```

```
Query Processed in 12.68 seconds.

Ended Executing this Stream at Fri Oct 24
05:56:21 2003

Stream Started at 1066946168.53
Stream Ended at 1066946181.21
Stream Processed in 12.68 seconds

SQL statements processed: 1
```

D.17 17.log

```
Begin Execution at Fri Oct 24 05:56:21 2003

-- using default substitutions

select
sum(l_extendedprice) / 7.0 as avg_yearly
from
lineitem,
part
where
p_partkey = l_partkey
and p_brand = 'Brand#23'
and p_container = 'MED BOX'
and l_quantity < (
select
0.2 * avg(l_quantity)
from
lineitem
where
l_partkey = p_partkey
)
```

```
                                        lineitem
AVG_YEARLY                              where
348406.05                               o_orderkey in (
                                        select
                                        l_orderkey
1 row  processed.                       from
Query Processed in 15.52 seconds.       lineitem
                                        group by
                                        l_orderkey having
Ended Executing this Stream at Fri Oct 24 sum(l_quantity) > 300
05:56:36 2003                           )
                                        and c_custkey = o_custkey
                                        and o_orderkey = l_orderkey
Stream Started at 1066946181.25         group by
Stream Ended at 1066946196.77           c_name,
Stream Processed in 15.52 seconds       c_custkey,
                                        o_orderkey,
                                        o_orderdate,
SQL statements processed: 1             o_totalprice
                                        order by
                                        o_totalprice desc,
                                        o_orderdate
                                        )
```

D.18 18.log

```
                                        where rownum <= 100

                                        C_NAME
Begin Execution at Fri Oct 24 05:56:36 2003 C_CUSTKEY                O_ORDERKEY
                                        O_ORDERDATE
                                        O_TOTALPRICE
                                        SUM(L_QUANTITY)
-- using default substitutions          Customer#000128120                128120.00
                                        4722021.00      1994-04-07
                                        544089.09       323.00
select * from (                         Customer#000144617                144617.00
select                                  3043270.00      1997-02-12
c_name,                                 530604.44       317.00
c_custkey,                              Customer#000013940                13940.00
o_orderkey,                             2232932.00      1997-04-13
o_orderdate,                            522720.61       304.00
o_totalprice,                           Customer#000066790                66790.00
sum(l_quantity)                         2199712.00      1996-09-30
from                                    515531.82       327.00
customer,                               Customer#000046435                46435.00
orders,                                 4745607.00      1997-07-03
```

508047.99	309.00		5296167.00	1996-09-06	
Customer#000015272		15272.00	469360.57	303.00	
3883783.00	1993-07-28		Customer#000147197		147197.00
500241.33	302.00		1263015.00	1997-02-02	
Customer#000146608		146608.00	467149.67	320.00	
3342468.00	1994-06-12		Customer#000064483		64483.00
499794.58	303.00		2745894.00	1996-07-04	
Customer#000096103		96103.00	466991.35	304.00	
5984582.00	1992-03-16		Customer#000136573		136573.00
494398.79	312.00		2761378.00	1996-05-31	
Customer#000024341		24341.00	461282.73	301.00	
1474818.00	1992-11-15		Customer#000016384		16384.00
491348.26	302.00		502886.00	1994-04-12	
Customer#000137446		137446.00	458378.92	312.00	
5489475.00	1997-05-23		Customer#000117919		117919.00
487763.25	311.00		2869152.00	1996-06-20	
Customer#000107590		107590.00	456815.92	317.00	
4267751.00	1994-11-04		Customer#000012251		12251.00
485141.38	301.00		735366.00	1993-11-24	
Customer#000050008		50008.00	455107.26	309.00	
2366755.00	1996-12-09		Customer#000120098		120098.00
483891.26	302.00		1971680.00	1995-06-14	
Customer#000015619		15619.00	453451.23	308.00	
3767271.00	1996-08-07		Customer#000066098		66098.00
480083.96	318.00		5007490.00	1992-08-07	
Customer#000077260		77260.00	453436.16	304.00	
1436544.00	1992-09-12		Customer#000117076		117076.00
479499.43	307.00		4290656.00	1997-02-05	
Customer#000109379		109379.00	449545.85	301.00	
5746311.00	1996-10-10		Customer#000129379		129379.00
478064.11	302.00		4720454.00	1997-06-07	
Customer#000054602		54602.00	448665.79	303.00	
5832321.00	1997-02-09		Customer#000126865		126865.00
471220.08	307.00		4702759.00	1994-11-07	
Customer#000105995		105995.00	447606.65	320.00	
2096705.00	1994-07-03		Customer#000088876		88876.00
469692.58	307.00		983201.00	1993-12-30	
Customer#000148885		148885.00	446717.46	304.00	
2942469.00	1992-05-31		Customer#000036619		36619.00
469630.44	313.00		4806726.00	1995-01-17	
Customer#000114586		114586.00	446704.09	328.00	
551136.00	1993-05-19		Customer#000141823		141823.00
469605.59	308.00		2806245.00	1996-12-29	
Customer#000105260		105260.00	446269.12	310.00	

Customer#000053029 53029.00 415200.61 304.00
2662214.00 1993-08-13 Customer#000105410 105410.00
446144.49 302.00 4478371.00 1996-03-05
Customer#000018188 18188.00 412754.51 302.00
3037414.00 1995-01-25 Customer#000149842 149842.00
443807.22 308.00 5156581.00 1994-05-30
Customer#000066533 66533.00 411329.35 302.00
29158.00 1995-10-21 Customer#000010129 10129.00
443576.50 305.00 5849444.00 1994-03-21
Customer#000037729 37729.00 409129.85 309.00
4134341.00 1995-06-29 Customer#000069904 69904.00
441082.97 309.00 1742403.00 1996-10-19
Customer#000003566 3566.00 408513.00 305.00
2329187.00 1998-01-04 Customer#000017746 17746.00
439803.36 304.00 6882.00 1997-04-09
Customer#000045538 45538.00 408446.93 303.00
4527553.00 1994-05-22 Customer#000013072 13072.00
436275.31 305.00 1481925.00 1998-03-15
Customer#000081581 81581.00 399195.47 301.00
4739650.00 1995-11-04 Customer#000082441 82441.00
435405.90 305.00 857959.00 1994-02-07
Customer#000119989 119989.00 382579.74 305.00
1544643.00 1997-09-20 Customer#000088703 88703.00
434568.25 320.00 2995076.00 1994-01-30
Customer#000003680 3680.00 363812.12 302.00
3861123.00 1998-07-03
433525.97 301.00
Customer#000113131 113131.00 57 rows processed.
967334.00 1995-12-15 Query Processed in 12.50 seconds.
432957.75 301.00
Customer#000141098 141098.00
565574.00 1995-09-24 Ended Executing this Stream at Fri Oct 24
430986.69 301.00 05:56:49 2003
Customer#000093392 93392.00
5200102.00 1997-01-22
425487.51 304.00 Stream Started at 1066946196.81
Customer#000015631 15631.00 Stream Ended at 1066946209.31
1845057.00 1994-05-12 Stream Processed in 12.50 seconds
419879.59 302.00
Customer#000112987 112987.00
4439686.00 1996-09-17 SQL statements processed: 1
418161.49 305.00
Customer#000012599 12599.00
4259524.00 1998-02-12

D.19 19.log

```
Begin Execution at Fri Oct 24 05:56:49 2003


-- using default substitutions


select
sum(l_extendedprice* (1 - l_discount)) as revenue
from
lineitem,
part
where
(
p_partkey = l_partkey
and p_brand = 'Brand#12'
and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
and l_quantity >= 1 and l_quantity <= 1 + 10
and p_size between 1 and 5
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
)
or
(
p_partkey = l_partkey
and p_brand = 'Brand#23'
and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
and l_quantity >= 10 and l_quantity <= 10 + 10
and p_size between 1 and 10
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
)
or
(
p_partkey = l_partkey
and p_brand = 'Brand#34'
and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
and l_quantity >= 20 and l_quantity <= 20 + 10
and p_size between 1 and 15
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
)


REVENUE
3083843.06


1 row  processed.
Query Processed in 7.20 seconds.


Ended Executing this Stream at Fri Oct 24 05:56:56 2003


Stream Started at 1066946209.35
Stream Ended at 1066946216.55
Stream Processed in 7.20 seconds


SQL statements processed: 1
```

D.20 20.log

```
Begin Execution at Fri Oct 24 05:56:56 2003


-- using default substitutions


select
s_name,
s_address
from
supplier,
```

```
nation
where
s_suppkey in (
select
ps_suppkey
from
partsupp
where
ps_partkey in (
select
p_partkey
from
part
where
p_name like 'forest%'
)
and ps_availqty > (
select
0.5 * sum(l_quantity)
from
lineitem
where
l_partkey = ps_partkey
and l_suppkey = ps_suppkey
and 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 s_nationkey = n_nationkey
and n_name = 'CANADA'
order by
s_name
```

```
S_NAME                    S_ADDRESS
Supplier#000000020
iybAE,RmTymrZVYaFZva2SH,j
Supplier#000000091
YV45D7TkfdQanOOZ7q9QxkyGUapU1oOWU
6q3
Supplier#000000197
YC2Acon6kjY3zj3Fbxs2k4Vdf7X0cd2F
Supplier#000000226
83qOdU2EYRdPQAQhEtn GRZEd
Supplier#000000285
Br7e1nnt1yxrw6ImgpJ7YdhFDjuBf
Supplier#000000378
FfbhyCxWvcPrO8ltp9
Supplier#000000402
i9Sw4DoyMhzhKXCH9By,AYSgmD
Supplier#000000530 0qwCMwobKY
OcmLyfRXlagA8ukENJv,
Supplier#000000688 D
fw5ocppmZpYBBIPI718hCihLDZ5KhKX
Supplier#000000710 f19YPvOyb
QoYwjKC,oPycpGfieBAcwKJo
Supplier#000000736
l6i2nMwVuovfKnuVgaSGK2rDy65DlAFLegiL
7
Supplier#000000761
zlSLelQUj2XrvTTFnv7WAcYZGvvMTx882d4
Supplier#000000884 bmhEShejaS
Supplier#000000887
urEaTejH5POADP2ARrf
Supplier#000000935 ij98czM
2KzWe7dDTOxB8sq0UfCdvrX
Supplier#000000975 ,AC
e,tBpNwKb5xMUzeohxlRn, hdZJo73gFQF8y
Supplier#000001263
rQWr6nf8ZhB2TAiIDIvo5Io
Supplier#000001399
LmrocnIMSyYOWuANx7
Supplier#000001446
lch9HMNU1R7a0LIybsUodVknk6
Supplier#000001454
TOpimgu2TVXIjhiL93h,
Supplier#000001500
wDmF5xLxtQch9ctVu,
Supplier#000001602 uKNWIeafaM644
Supplier#000001626
UhxNRzUu1dtFmp0
Supplier#000001682
pXTkGxrTQVyH1Rr
Supplier#000001699
Q9C4rfJ26oijVPqqcqVXeRI
Supplier#000001700
7hMlCof1Y5zLFg
```

Supplier#000001726
TeRY7TtTH24sEword7yAaSkjx8
Supplier#000001730 Rc8e,1Pybn
r6zo0VJIEiD0UD vhk
Supplier#000001746
qWsendlOekQG1aW4uq06uQaCm51se8lirv7
hBRd
Supplier#000001752
Fra7outx41THYJaRThdOGiBk
Supplier#000001856
jXcRgzYF0ah05iR8p6w5SbJJLcUGyYiURPvF
wUWM
Supplier#000001931
FpJbMU2h6ZR2eBv8I9NIxF
Supplier#000001939 Nrk,JA4bfReUs
Supplier#000001990
DSDJkCgBJzuPg1yuM,CUdLnsRliOxkkHezTC
A
Supplier#000002020 jB6r1d7MxP6co
Supplier#000002022
dwebGX7Id2pc25YvY33
Supplier#000002036
20ytTtVObjKUUI2WCB0A
Supplier#000002204
uYmlr46C06udCqanj0KiRsoTQakZsEyssL
Supplier#000002243 nSOEV3JeOU79
Supplier#000002245
hz2qWXWVjOyKhqPYMoEwz6zFkrTaDM
Supplier#000002282
ES21K9dxoW1I1TzWCj7ekdlNwSWnv1Z
6mQ,BKn
Supplier#000002303
nCoWfpB6YOymbgOht7ltfklpkHl
Supplier#000002373
RzHSxOTQmElCjxIBiVA52Z JB58rJhPRylR
Supplier#000002419
qydBQd14I5l5mVXa4fYY
Supplier#000002481
nLKHUOn2Ml9TOA06Znq9GEMcIlMO2
Supplier#000002571 JZUugz04c
iJFLrlGsz9O N,W 1rVHNIReyq
Supplier#000002585
CsPoKpw2QuTY4AV1NkWuttneIa4SN
Supplier#000002630
ZIQAvjNUY9KH5ive zm7k VlPiDl7CCo21
Supplier#000002719
4nnzQI2CbqREQUuIsXTBVUkaP4mNS3
Supplier#000002721
HVdFAN2JHMQSpKm
Supplier#000002730
lIFxR4fzm31C6,muzJwl84z
Supplier#000002775 yDclaDaBD4ihH
Supplier#000002853 rTNAOItXka
Supplier#000002875 6JgMi
9Qt6VmwL3Ltt1SRlKww0keLQ,RAZa
Supplier#000002934
m,trBENywSArwg3DhB
Supplier#000002941 Naddba
8YTEKekZyP0
Supplier#000002960
KCPCEsRGGo6vx8TygHh60nAYf9rStQT2T
Supplier#000002980
B9k9yVsyaXvWktOSHezqHiAEp9id0SKzkw
Supplier#000003062
LSQNgqY1xnOzz9zBCapy7HwOZQ
Supplier#000003087
ANwe8QsZ4rgj1HSqVz991eWQ
Supplier#000003089 s5b
VCIZqMSZVa r g7LTdcg29GbTE7rI1x
Supplier#000003095
HxON3jJhUi3zjt,r mTD
Supplier#000003201
E87yws6I,t0qNs4QW7UzExKiJnJDZWue
Supplier#000003213
pxrRP4irQ1VoyfQ,dTf3
Supplier#000003241
j06SU,LS9O3mwjAMOViANeIhb
Supplier#000003275
9xO4nyJ2QJcX6vGf
Supplier#000003288
EDdfNt7E5Uc,xLTupoIgYL4yY7ujh,
Supplier#000003313
El2I7we,049SPrvomUm4hZwJoOhZkvLxLJXg
VH
Supplier#000003314
jnisU8MzqO4iUB3zsPcrysMw3DDUojS4q7LD
Supplier#000003380
jPv0V,pszouuFT3YsAqlP,kxT3u,gTFiEbRt,x

Supplier#000003403
e3X2o ,KCG9tsHji8A XXCxiF2hZWBw
Supplier#000003421
Sh3dt9W5oeofFWovnFhrg,
Supplier#000003441
zvFJIzS,oUuShHjpcX
Supplier#000003590
sy79CMLxqb,Cbo
Supplier#000003607
lNqFHQYjwSAkf
Supplier#000003625
qY588W0Yk5iaUy1RXTgNrEKrMAjBYHcKs
Supplier#000003656 eEYmmO2gmD
JdfG32XtDgJV,db56
Supplier#000003782
iVsPZg7bk06TqNMwi0LKbLUrC1zmrg
Supplier#000003918
meRvRCsJoAbfqd0Re4
Supplier#000003941
Pmb05mQfBMS618O7WKqZJ 9vyv
Supplier#000003994 W00LZp3NjK0
Supplier#000004005
V723F1wCy2eA4OgIu8TjBtOVUHp
Supplier#000004033
ncsAhv9Je,kFXTNjfb2
Supplier#000004140 0hL7DJyYjcHL
Supplier#000004165
wTJ2dZNQA8P2oi99N6DT47ndHy,XKD2
Supplier#000004207
tF64pwiOM4IkWjN3mS,e06WuAjLx
Supplier#000004236
dl,HPtJmGipxYsSqn9wmqkuWjst,mCeJ8O6T
Supplier#000004246 Xha
aXQF7u4qU3LsHD
Supplier#000004278 bBddbpBxIVp
Di9
Supplier#000004343
GK3sbopqrQEkWLMvVBFCG
Supplier#000004346 S3076LEOwo
Supplier#000004388 VfZ l1J,mwp4aS
Supplier#000004406
Ah0ZaLu6VwufPWUz,7kbXgYZhauEaHqGIg
Supplier#000004430
yvSsKNSTL5HLXBET4luOsPNLxKzAMk
Supplier#000004522
xXtCKwsZDArxIBGDfzX2PgobGZsBg
Supplier#000004527 p
pVXCnxgcklWF6A1o3OHY3qW6
Supplier#000004542
NJSbLJDroYG2y1r3rDiKg
Supplier#000004574
1HvGwnVueZ5CIndc
Supplier#000004655 67NqBc4
t3PG3F8aO IsqWNq4kGaPowYL
Supplier#000004701
6jX4u47URzIMHf
Supplier#000004711 bEzjp1QdQu
ls2ERMxv0km vn6bu2zXlL1
Supplier#000004987
UFx1upJ8MvOvgFjA8
Supplier#000005000 DeX804
w0H8FrCUvahgy ilbuzBX3NK
Supplier#000005100
OfvYPs3Io,wEvvLHNaLuCX
Supplier#000005192
JDp4rhXiDw0kf6RH
Supplier#000005195 Woi3b2ZaicPh
ZSfu1EfXhE
Supplier#000005283
5fxYXxwXy,TQX,MqDC2hxzyQ
Supplier#000005300 gXG28YqpxU
Supplier#000005386
Ub6AAfHpWLWP
Supplier#000005426 9Dz2OVT1q
sb4BK71ljQ1XjPBYRPvO
Supplier#000005484 saFdOR
qW7AFY,3asPqiiAa11Mo22pCoN0BtPrKo
Supplier#000005505
d2sbjG43KwMPX
Supplier#000005506 On f5ypzoWgB
Supplier#000005516
XsN99Ks9wEvcohU6jRD2MeebQFf76mD8vov
uY
Supplier#000005536
Nzo9tGkpgbHT,EZ4D,77MYKl4ah1C
Supplier#000005605
7Vj6Eil0mThqkM
Supplier#000005631

14TVrjlzo2SJEBYCDgpMwTlvwSqC
Supplier#000005730 5rkb0PSews HvxkL8JaD41UpnSF2cg8H1
Supplier#000005736 2dq XTYhtYWSfp
Supplier#000005737 dmEWcS32C3kx,d,B95 OmYn48
Supplier#000005797 ,o,OebwRbSDm Vl9gN9fpWPCiqB UogvlSR
Supplier#000005836 tx3SjPD2ZuWGFBRH,
Supplier#000005875 lK,sYiGzB94hSyHy9xvSZFbVQNCZe2LXZuG bS
Supplier#000005974 REhR5jE,lLusQXvf54SwYySgsSSVFhu
Supplier#000005989 rjFY,5kgLpBu7c
Supplier#000006059 4m0cv8MwJ9yX2vlwI Z
Supplier#000006065 UiI2Cy3W4Tu5sLk LuvXLRy6KihlGv
Supplier#000006070 TalC5m0pDrO6DZbngfmGmqe
Supplier#000006109 rY5gbfh3dKHnylcQUTPGCwnbe
Supplier#000006121 S92ycWwEzYYw4GspCBJN1WMuHhoZ
Supplier#000006215 j2iEbTsl,5PWdqWZ7k1yiISb7qtiiZljDIPEo
Supplier#000006217 RVN23SYT9jenUeaWGXUd
Supplier#000006274 S3yTZWqxTKUq g QQgcW9 AqhCkNZsW51hHuwU
Supplier#000006435 xIgE69XszYbnO4Eon7cHHO8y
Supplier#000006463 7 wkdj2EO49iotley2kmIM ADpLSszGV3RNWj
Supplier#000006493 ojV f,sNaB6Hm7r,fknDVTL63raJgAjZK
Supplier#000006521 b9 2zjHzxR
Supplier#000006607 3F 2e2gqD5u5B
Supplier#000006706 Ak4ga,ePu1QZ6C3qkrqjosaX0gxvqS9vkbe
Supplier#000006761 n4jhxGMqB5prD1HhpLvwrWStOLlla
Supplier#000006808 HGd2Xo 9nEcHJhZvXjXxWKIpApT
Supplier#000006858 fnlINT885vBBhsWwTGiZ0o22thwGY16h GHJj21
Supplier#000006872 XIDPiA7PLXCWK6SeEcld
Supplier#000006949 mLxYUJhsGcLtKe ,GFirNu183AvT
Supplier#000006985 PrUUiboQpy,OtgJ01Z4BxJQUyrw9c3I
Supplier#000007072 2tRyX9M1a 4Rcm57s779F1ANG9jlpK
Supplier#000007098 G3j8g0KC4OcbAu2OVoPHrXQWMCUdjq8wg CHOExu
Supplier#000007135 ls DoKV7V5ulfQy9V
Supplier#000007160 TqDGBULB3cTqIT6FKDvm9BS4e4v,zwYiQP b
Supplier#000007169 tEc95D2moN9S84nd55O,dlnW
Supplier#000007322 wr7dgte5q MAjiY0uwmi3MyDkSMX1
Supplier#000007365 51xhROLvQMJ05DndtZWt
Supplier#000007398 V8eE6oZ00OFNU,
Supplier#000007402 4UVv58ery1rjmqSR5
Supplier#000007448 yhhpWiJi7EJ6Q5VCaQ
Supplier#000007477 9m9j0wfhWzCvVHxkU,PpAxwSH0h
Supplier#000007509 q8,V6LJRoHJjHcOuSG7aLTMg
Supplier#000007561 rMcFg2530VC
Supplier#000007789 rQ7cUcPrtudOyO3svNSkimqH6qrfWT2Sz
Supplier#000007801 69fi,U1r6enUb
Supplier#000007818 yhhc2CQec Jrvc8zqBi83

Supplier#000007885
u3sicchh5ZpyTUpN1cJKNcAoabIWgY
Supplier#000007918
r,v9mBQ6LoEYyj1
Supplier#000007926 ErzCF80K9Uy
Supplier#000007957 ELwnio14ssoU1
dRyZIL OK3Vtzb
Supplier#000007965 F7Un5lJ7p5hhj
Supplier#000007968
DsF9UlZ2Fo6HXN9aErvyg1ikHoD582HSGZp
P
Supplier#000007998
LnASFBfYRFOo9d6d,asBvVq9Lo2P
Supplier#000008168
aOa82a8ZbKCnfDLX
Supplier#000008231 IK7eGw
Yj90sTdpsP,vcqWxLB
Supplier#000008243
2AyePMkDqmzVzjGTizXthFLo8h
EiudCMxOmIIG
Supplier#000008275
BlbNDfWg,gpXKQlLN
Supplier#000008323 75I18sZmASwm
POeheRMdj9tmpyeQ,BfCXN5BIAb
Supplier#000008366
h778cEj14BuW9OEKlvPTWq4iwASR6EBBX
N7zeS8
Supplier#000008423
RQhKnkAhR0DAr3Ix4Q1weMMn00hNe Kq
Supplier#000008480
4sSDA4ACReklNjEm5T6b
Supplier#000008532
Uc29q4,5xVdDOF87UZrxhr4xWS0ihEUXuh
Supplier#000008595 MH0iB73GQ3z
UW3O DbCbqmc
Supplier#000008610
SgVgP90vP452sUNTgzL9zKwXHXAzV6tV
Supplier#000008705
aE,trRNdPx,4yinTD9O3DebDIp
Supplier#000008742
HmPlQEzKCPEcTUL14,kKq
Supplier#000008841 I
85Lu1sekbg2xrSIzm0
Supplier#000008895
2cH4okfaLSZTTg8sKRbbJQxkmeFu2Esj
Supplier#000008967 2kwEHyMG
7FwozNImAUE6mH0hYtqYculJM
Supplier#000008972 w2vF6
D5YZO3visPXsqVfLADTK
Supplier#000009032
qK,trB6Sdy4Dz1BRUFNy
Supplier#000009147
rOAuryHxpZ9eOvx
Supplier#000009252 F7cZaPUHwh1
ZKyj3xmAVWC1XdP ue1p5m,i
Supplier#000009278 RqYTzgxj93CLX
0mcYfCENOefD
Supplier#000009327
uoqMdf7e7Gj9dbQ53
Supplier#000009430 igRqmneFt
Supplier#000009567
r4Wfx4c3xsEAjcGj71HHZByornl D9vrztXlv4
Supplier#000009601
51m637bO,Rw5DnHWFUvLacRx9
Supplier#000009709
rRnCbHYgDgl9PZYnyWKVYSUW0vKg
Supplier#000009753
wLhVEcRmd7PkJF4FBnGK7Z
Supplier#000009796
z,y4Idmr15DOvPUqYG
Supplier#000009799
4wNjXGa4OKWl
Supplier#000009811
E3iuyq7UnZxU7oPZIe2Gu6
Supplier#000009812
APFRMy3lCbgFga53n5t9DxzFPQPgnjrGt32
Supplier#000009862 rJzweWeN58
Supplier#000009868
ROjGgx5gvtkmnUUoeyy7v
Supplier#000009869
ucLqxzrpBTRMewGSM29t0rNTM30g1Tu3Xgg
3mKag
Supplier#000009899
7XdpAHrzr1t,UQFZE
Supplier#000009974
7wJ,J5DKcxSU4Kp1cQLpbcAvB5AsvKT

```
204 rows processed.
Query Processed in 8.83 seconds.

Ended Executing this Stream at Fri Oct 24 05:57:05 2003

Stream Started at 1066946216.58
Stream Ended at 1066946225.42
Stream Processed in 8.83 seconds

SQL statements processed: 1
```

D.21 21.log

```
Begin Execution at Fri Oct 24 05:57:05 2003

-- using default substitutions

select * from (
select
s_name,
count(*)    numwait
from
supplier,
lineitem l1,
orders,
nation
where
s_suppkey = l1.l_suppkey
and o_orderkey = l1.l_orderkey
and o_orderstatus = 'F'
and l1.l_receiptdate > l1.l_commitdate
and exists (
select
*
from
lineitem l2
where
l2.l_orderkey = l1.l_orderkey
and l2.l_suppkey <> l1.l_suppkey
)
and not exists (
select
*
from
lineitem l3
where
l3.l_orderkey = l1.l_orderkey
and l3.l_suppkey <> l1.l_suppkey
and l3.l_receiptdate > l3.l_commitdate
)
and s_nationkey = n_nationkey
and n_name = 'SAUDI ARABIA'
group by
s_name
order by
numwait desc,
s_name)
where rownum <= 100

S_NAME                    NUMWAIT
Supplier#000002829        20.00
Supplier#000005808        18.00
Supplier#000000262        17.00
Supplier#000000496        17.00
Supplier#000002160        17.00
Supplier#000002301        17.00
Supplier#000002540        17.00
Supplier#000003063        17.00
Supplier#000005178        17.00
Supplier#000008331        17.00
Supplier#000002005        16.00
Supplier#000002095        16.00
Supplier#000005799        16.00
Supplier#000005842        16.00
Supplier#000006450        16.00
Supplier#000006939        16.00
Supplier#000009200        16.00
```

Supplier#000009727	16.00	Supplier#000002615	13.00
Supplier#000000486	15.00	Supplier#000002978	13.00
Supplier#000000565	15.00	Supplier#000003048	13.00
Supplier#000001046	15.00	Supplier#000003234	13.00
Supplier#000001047	15.00	Supplier#000003727	13.00
Supplier#000001161	15.00	Supplier#000003806	13.00
Supplier#000001336	15.00	Supplier#000004472	13.00
Supplier#000001435	15.00	Supplier#000005236	13.00
Supplier#000003075	15.00	Supplier#000005906	13.00
Supplier#000003335	15.00	Supplier#000006241	13.00
Supplier#000005649	15.00	Supplier#000006326	13.00
Supplier#000006027	15.00	Supplier#000006384	13.00
Supplier#000006795	15.00	Supplier#000006394	13.00
Supplier#000006800	15.00	Supplier#000006624	13.00
Supplier#000006824	15.00	Supplier#000006629	13.00
Supplier#000007131	15.00	Supplier#000006682	13.00
Supplier#000007382	15.00	Supplier#000006737	13.00
Supplier#000008913	15.00	Supplier#000006825	13.00
Supplier#000009787	15.00	Supplier#000007021	13.00
Supplier#000000633	14.00	Supplier#000007417	13.00
Supplier#000001960	14.00	Supplier#000007497	13.00
Supplier#000002323	14.00	Supplier#000007602	13.00
Supplier#000002490	14.00	Supplier#000008134	13.00
Supplier#000002993	14.00	Supplier#000008234	13.00
Supplier#000003101	14.00	Supplier#000009435	13.00
Supplier#000004489	14.00	Supplier#000009436	13.00
Supplier#000005435	14.00	Supplier#000009564	13.00
Supplier#000005583	14.00	Supplier#000009896	13.00
Supplier#000005774	14.00	Supplier#000000379	12.00
Supplier#000007579	14.00	Supplier#000000673	12.00
Supplier#000008180	14.00	Supplier#000000762	12.00
Supplier#000008695	14.00	Supplier#000000811	12.00
Supplier#000009224	14.00	Supplier#000000821	12.00
Supplier#000000357	13.00	Supplier#000001337	12.00
Supplier#000000436	13.00	Supplier#000001916	12.00
Supplier#000000610	13.00	Supplier#000001925	12.00
Supplier#000000788	13.00	Supplier#000002039	12.00
Supplier#000000889	13.00	Supplier#000002357	12.00
Supplier#000001062	13.00	Supplier#000002483	12.00
Supplier#000001498	13.00		
Supplier#000002056	13.00		
Supplier#000002312	13.00	100 rows processed.	
Supplier#000002344	13.00	Query Processed in 24.91 seconds.	
Supplier#000002596	13.00		

```
Ended Executing this Stream at Fri Oct 24 05:57:30 2003

Stream Started at 1066946225.46
Stream Ended at 1066946250.36
Stream Processed in 24.91 seconds

SQL statements processed: 1
```

D.22 22.log

```
Begin Execution at Fri Oct 24 05:57:30 2003

-- using default substitutions

select
cntrycode,
count(*) as numcust,
sum(c_acctbal) as totacctbal
from
(
select
substr(c_phone, 1, 2) as cntrycode,
c_acctbal
from
customer
where
substr(c_phone,1, 2) in
('13', '31', '23', '29', '30', '18', '17')
and c_acctbal > (
select
avg(c_acctbal)
from
customer
where
c_acctbal > 0.00
and substr(c_phone, 1, 2) in
('13', '31', '23', '29', '30', '18', '17')
)
and not exists (
select
*
from
orders
where
o_custkey = c_custkey
)
) custsale
group by
cntrycode
order by
cntrycode

CNTRYCODE                NUMCUST
TOTACCTBAL
13                           888.00
6737713.99
17                           861.00
6460573.72
18                           964.00
7236687.40
23                           892.00
6701457.95
29                           948.00
7158866.63
30                           909.00
6808436.13
31                           922.00
6806670.18

7 rows processed.
Query Processed in 8.01 seconds.

Ended Executing this Stream at Fri Oct 24 05:57:38 2003
```

Stream Started at 1066946250.40
Stream Ended at 1066946258.41
Stream Processed in 8.01 seconds

SQL statements processed: 1

Appendix E Seed and Input Parameters

E.1 seed

1028120446

E.2 stream00

14 1993-05-01
2 43 STEEL EUROPE
9 beige
20 dim 1994-01-01 CHINA
6 1997-01-01 0.05 25
17 Brand#45 SM BOX
18 315
8 CHINA ASIA STANDARD BURNISHED BRASS
21 INDIA
13 unusual packages
3 AUTOMOBILE 1995-03-24
22 21 12 23 30 25 22 10
16 Brand#14 PROMO ANODIZED 18 13 17 25 23 44 42 4
4 1996-09-01
11 BRAZIL 0.0000001000
15 1993-03-01
1 85
10 1994-12-01
19 Brand#12 Brand#24 Brand#55 9 15 21
5 AMERICA 1997-01-01
7 JAPAN CHINA
12 AIR MAIL 1993-01-01

E.3 stream01

21 ALGERIA
3 HOUSEHOLD 1995-03-09
18 312
5 ASIA 1997-01-01
11 MOROCCO 0.0000001000
7 CANADA IRAN
6 1997-01-01 0.03 24
20 peach 1993-01-01 INDIA
17 Brand#42 SM PACK
12 REG AIR FOB 1993-01-01
16 Brand#44 SMALL PLATED 3 2 15 45 44 28 18 12
15 1995-09-01
13 unusual packages
10 1993-10-01
2 31 BRASS AMERICA
8 IRAN MIDDLE EAST PROMO BRUSHED BRASS
14 1993-09-01
19 Brand#14 Brand#12 Brand#54 4 16 29
9 white
22 18 21 27 13 20 14 26
1 93
4 1994-05-01

E.4 stream02

6 1997-01-01 0.08 24
17 Brand#44 SM DRUM
14 1993-12-01
16 Brand#34 LARGE POLISHED 10 13 36 12 15 28 1 35
19 Brand#11 Brand#45 Brand#43 9 17 25
10 1994-07-01
9 tan
2 19 NICKEL MIDDLE EAST
15 1993-06-01
8 BRAZIL AMERICA PROMO PLATED BRASS
5 EUROPE 1997-01-01
22 19 18 15 16 27 21 31
12 SHIP FOB 1993-01-01

7 SAUDI ARABIA BRAZIL
13 express packages
18 314
1 101
4 1996-12-01
20 blue 1996-01-01 RUSSIA
3 AUTOMOBILE 1995-03-26
11 CANADA 0.0000001000
21 PERU

E.5 stream03

8 ROMANIA EUROPE PROMO ANODIZED BRASS
5 MIDDLE EAST 1997-01-01
4 1994-09-01
6 1997-01-01 0.06 25
17 Brand#41 LG BOX
7 JAPAN ROMANIA
1 109
18 312
22 22 28 19 21 10 20 23
14 1994-03-01
9 sky
10 1993-04-01
15 1996-01-01
11 MOZAMBIQUE 0.0000001000
20 linen 1994-01-01 JAPAN
2 7 TIN AMERICA
21 INDONESIA
19 Brand#23 Brand#23 Brand#42 5 18 21
13 express packages
16 Brand#14 STANDARD ANODIZED 13 26 20 43 16 3 9 15
12 FOB AIR 1996-01-01
3 FURNITURE 1995-03-11

E.6 stream04

11 PERU 0.0000001000
13 express requests
3 FURNITURE 1995-03-13
1 64
5 AFRICA 1997-01-01
21 ARGENTINA
14 1994-06-01
19 Brand#25 Brand#11 Brand#42 10 19 28
15 1993-09-01
17 Brand#42 LG PACK
12 MAIL FOB 1994-01-01
6 1997-01-01 0.03 24
4 1997-04-01
9 royal
8 IRAQ MIDDLE EAST ECONOMY POLISHED BRASS
16 Brand#44 MEDIUM BURNISHED 16 12 4 11 33 18 2 29
11 EGYPT 0.0000001000
2 44 STEEL MIDDLE EAST
10 1994-01-01
18 313
1 117
13 express requests
7 EGYPT IRAQ
22 12 14 24 33 21 28 18
3 AUTOMOBILE 1995-03-28
20 thistle 1993-01-01 BRAZIL

E.7 stream05

21 ROMANIA
15 1996-04-01
4 1995-01-01
6 1993-01-01 0.08 24
7 VIETNAM CANADA
16 Brand#34 ECONOMY POLISHED 20 5 39 35 43 2 23 47
19 Brand#22 Brand#44 Brand#31 5 20 24
18 315
14 1994-09-01
22 27 24 12 22 30 13 14
2 32 BRASS ASIA
5 AMERICA 1993-01-01
8 CANADA AMERICA ECONOMY BURNISHED STEEL

20 ghost 1996-01-01 PERU
12 RAIL SHIP 1994-01-01
17 Brand#44 LG DRUM
10 1994-10-01
9 powder

E.8 stream06

10 1993-08-01
3 MACHINERY 1995-03-30
15 1994-01-01
13 express requests
6 1993-01-01 0.06 25
8 SAUDI ARABIA MIDDLE EAST LARGE BRUSHED STEEL
9 pale
7 JORDAN SAUDI ARABIA
4 1997-08-01
11 ETHIOPIA 0.0000001000
22 16 19 15 34 22 33 25
18 312
12 AIR SHIP 1994-01-01
1 72
5 EUROPE 1993-01-01
16 Brand#14 STANDARD BRUSHED 23 28 7 48 9 47 50 22
2 20 NICKEL MIDDLE EAST
14 1995-01-01
19 Brand#34 Brand#32 Brand#35 10 10 20
20 rose 1995-01-01 FRANCE
17 Brand#41 MED BAG
21 IRAQ

E.9 stream07

18 314
8 JAPAN ASIA LARGE PLATED STEEL
20 coral 1993-01-01 VIETNAM
21 CANADA
2 8 TIN ASIA
4 1995-05-01
22 17 21 29 14 22 32 23
17 Brand#43 MED PACK
1 80
11 CHINA 0.0000001000
9 moccasin
19 Brand#31 Brand#15 Brand#34 6 11 28
3 FURNITURE 1995-03-15
13 express requests
5 MIDDLE EAST 1993-01-01
7 ETHIOPIA JAPAN
10 1994-05-01
16 Brand#44 LARGE BURNISHED 39 10 8 19 43 13 47 27
6 1993-01-01 0.03 24
14 1995-04-01
15 1996-08-01
12 REG AIR SHIP 1995-01-01

Appendix F Benchmark Scripts

F.1 dbtables.sql

```
set echo on
set numwidth 25
spool rdbtablest
SELECT COUNT(*) FROM LINEITEM;

SELECT * FROM LINEITEM
WHERE L_ORDERKEY IN
( 4, 26598, 148577, 387431, 56704,
517442, 600000)
AND L_LINENUMBER = 1
ORDER BY L_ORDERKEY;

SELECT * FROM REGION;

SELECT COUNT(*) FROM NATION;

SELECT * FROM NATION
WHERE N_NATIONKEY IN (3,10,14,20)
ORDER BY N_NATIONKEY;

SELECT COUNT(*) FROM ORDERS;

SELECT * FROM ORDERS
WHERE O_ORDERKEY IN ( 7, 44065,
287590, 411111, 483876, 599942 )
ORDER BY O_ORDERKEY;

SELECT COUNT(*) FROM PART;

SELECT * FROM PART
WHERE P_PARTKEY IN
(1,984,8743,9028,13876,17899,20000)
ORDER BY P_PARTKEY;

SELECT COUNT(*) FROM PARTSUPP;

SELECT* FROM PARTSUPP
  WHERE PS_PARTKEY = 3398
   AND PS_SUPPKEY = (SELECT
MIN(PS_SUPPKEY)
      FROM PARTSUPP WHERE
PS_PARTKEY = 3398);

SELECT* FROM PARTSUPP
  WHERE PS_PARTKEY =15873
   AND PS_SUPPKEY = (SELECT
MIN(PS_SUPPKEY)
      FROM PARTSUPP WHERE
PS_PARTKEY = 15873);

SELECT* FROM PARTSUPP
  WHERE PS_PARTKEY = 11394
   AND PS_SUPPKEY = (SELECT
MIN(PS_SUPPKEY)
      FROM PARTSUPP WHERE
PS_PARTKEY = 11394);

SELECT* FROM PARTSUPP
  WHERE PS_PARTKEY = 6743
   AND PS_SUPPKEY = (SELECT
MIN(PS_SUPPKEY)
      FROM PARTSUPP WHERE
PS_PARTKEY = 6743);

SELECT* FROM PARTSUPP
  WHERE PS_PARTKEY = 19763
   AND PS_SUPPKEY = (SELECT
MIN(PS_SUPPKEY)
```

```
        FROM PARTSUPP WHERE
PS_PARTKEY =19763);



SELECT COUNT(*) FROM SUPPLIER;

SELECT * FROM SUPPLIER
WHERE S_SUPPKEY IN
(83,265,492,784,901,1000)
ORDER BY S_SUPPKEY;



DROP TABLE MINMAX;

CREATE TABLE MINMAX
(TNAME CHAR(15),
KEYMIN INTEGER,
KEYMAX INTEGER);

INSERT INTO MINMAX
SELECT
'LINEITEM_ORD',MIN(L_ORDERKEY),MA
X(L_ORDERKEY)
FROM LINEITEM ;

INSERT INTO MINMAX
SELECT
'LINEITEM_NBR',MIN(L_LINENUMBER),
MAX(L_LINENUMBER)
FROM LINEITEM;

INSERT INTO MINMAX
SELECT
'ORDERTBL',MIN(O_ORDERKEY),MAX(O
_ORDERKEY)
FROM ORDERS;

INSERT INTO MINMAX
SELECT
'CUSTOMER',MIN(C_CUSTKEY),MAX(C_
CUSTKEY)
FROM CUSTOMER;

INSERT INTO MINMAX
SELECT
'PART',MIN(P_PARTKEY),MAX(P_PARTKE
Y)
FROM PART;

INSERT INTO MINMAX
SELECT
'SUPPLIER',MIN(S_SUPPKEY),MAX(S_SU
PPKEY)
FROM SUPPLIER;

INSERT INTO MINMAX
SELECT
'PARTSUPP_PART',MIN(PS_PARTKEY),MA
X(PS_PARTKEY)
FROM PARTSUPP;

INSERT INTO MINMAX
SELECT
'PARTSUPP_SUPP',MIN(PS_SUPPKEY),MA
X(PS_SUPPKEY)
FROM PARTSUPP ;

INSERT INTO MINMAX
SELECT
'NATION',MIN(N_NATIONKEY),MAX(N_N
ATIONKEY)
FROM NATION;

INSERT INTO MINMAX
SELECT
'REGION',MIN(R_REGIONKEY),MAX(R_R
EGIONKEY)
FROM REGION;

SELECT * FROM MINMAX;
spool off
exit;
```

F.2 dbinsert.sql

```
rem
```

```
rem
================================
================================
+
rem   FILENAME
rem          inserts.sql
rem   DESCRIPTION
rem          Inserts duplicate rows with new
key numbers and
rem          inserts rows with values beyond
the TPC-D values.
rem
rem
================================
================================
rem
rem Usage:   sqlplus tpcc/tpcc @insert
rem



set pagesize 100
set termout on
set echo on
set timing on
spool rdbinsert



rem
================================
================================
rem Duplicates
rem
================================
================================

REM get timestamp
select    to_char    (sysdate,    'HH24:MI:SS')
timestamp from dual;



drop table temp_part;
create table temp_part as
   select * from part
      where p_partkey = 1;

update temp_part
      set p_partkey = 2147483647;
insert into part
      (select * from temp_part);
select * from part
      where p_partkey = 2147483647
      or p_partkey = 1;
delete from part
      where p_partkey = 2147483647;
drop table temp_part;
commit;

drop table temp_supplier;
create table temp_supplier as
   select * from supplier
      where s_suppkey = 1;
update temp_supplier
      set s_suppkey = 2147483647;
insert into supplier
      (select * from temp_supplier);
select * from supplier
      where s_suppkey = 2147483647
      or s_suppkey = 1;
delete from supplier
      where s_suppkey = 2147483647;
drop table temp_supplier;
commit;

drop table temp_partsupp;
create table temp_partsupp as
   select * from partsupp
      where ps_partkey = 1
      and ps_suppkey = 2;
update temp_partsupp
      set ps_partkey = 2147483647,
          ps_suppkey = 2147483647;
insert into partsupp
      (select * from temp_partsupp);
select * from partsupp
      where (ps_partkey = 2147483647
      and    ps_suppkey = 2147483647)
      or    (ps_partkey = 1
      and    ps_suppkey = 2);
delete from partsupp
```

```
        where ps_partkey = 2147483647
        and    ps_suppkey = 2147483647;
drop table temp_partsupp;
commit;



drop table temp_customer;
create table temp_customer as
    select * from customer
        where c_custkey = 1;
update temp_customer
        set c_custkey = 2147483647;
insert into customer
        (select * from temp_customer);
select * from customer
        where c_custkey = 2147483647
        or c_custkey = 1;
delete from customer
        where c_custkey = 2147483647;
drop table temp_customer;
commit;

drop table temp_orders;
create table temp_orders as
   select * from orders
        where o_orderkey = (select
min(o_orderkey) from orders);
update temp_orders
        set o_orderkey = 2147483647;
insert into orders
        (select * from temp_orders);
select * from orders
        where o_orderkey = 2147483647
        or o_orderkey = (select min(o_orderkey)
from orders);
delete from orders
        where o_orderkey = 2147483647;
drop table temp_orders;
commit;

drop table temp_lineitem;
create table temp_lineitem as
   select * from lineitem
        where l_orderkey = (select
min(o_orderkey) from orders)
        and    l_linenumber = 1;
update temp_lineitem
        set l_orderkey = 2147483647,
            l_partkey   = 2147483647,
            l_suppkey   = 2147483647,
            l_linenumber = -2147483646;
insert into lineitem
        (select * from temp_lineitem);
select * from lineitem
        where (l_orderkey = 2147483647
        and     l_partkey   = 2147483647
        and     l_suppkey   = 2147483647
        and     l_linenumber = -2147483646)
        or    (l_orderkey          =          (select
min(o_orderkey) from orders)
        and     l_linenumber = 1);
delete from lineitem
        where l_orderkey = 2147483647
        and    l_partkey   = 2147483647
        and    l_suppkey   = 2147483647
        and    l_linenumber = -2147483646;
drop table temp_lineitem;
commit;

drop table temp_nation;
create table temp_nation as
   select * from nation
        where n_nationkey = 1;
update temp_nation
        set n_nationkey = 2147483647;
insert into nation
        (select * from temp_nation);
select * from nation
        where n_nationkey = 2147483647
        or n_nationkey = 1;
delete from nation
        where n_nationkey = 2147483647;
drop table temp_nation;
commit;


drop table temp_region;
create table temp_region as
```

```
   select * from region
      where r_regionkey = 1;
update temp_region
      set r_regionkey = 2147483647;
insert into region
      (select * from temp_region);
select * from region
      where r_regionkey = 2147483647
      or r_regionkey = 1;
delete from region
      where r_regionkey = 2147483647;
drop table temp_region;
commit;

rem
================================
================================
rem Duplicates finished starting inserts for
domain range
rem
================================
================================

REM get timestamp
select to_char (sysdate, 'HH24:MI:SS')
timestamp from dual;



insert into supplier
      (s_suppkey, s_name, s_address,
s_nationkey, s_phone,
      s_acctbal, s_comment)
      values
      (2147483647, 'NAME text .....25E',
      'Address varchar .....30.....40E',
      2147483647,'This is phone E',
123456789012,
      'Supplier comment field is 101 long no
E');
select * from supplier
      where s_suppkey = 2147483647;
delete from supplier D
      where s_suppkey = 2147483647;
rem
================================
================================
insert into part
      (p_partkey, p_name, p_mfgr, p_brand,
p_type,
      p_size, p_container, p_retailprice,
p_comment)
      values
      (2147483647, 'Pname
text .....2.....3.....4....5E',
      'Pmfgr text.....2....5E','Pbrand 10E',
      'Ptype varchar.....2....5E', 2147483646,
      'PcontainrE', 123456789012,
      'Part comment field  23E');
select * from part
      where p_partkey = 2147483647;
delete from part
      where p_partkey = 2147483647;
rem
================================
================================
insert into partsupp
      (ps_partkey, ps_suppkey, ps_availqty,
ps_supplycost,
      ps_comment)
      values
     (2147483647, 2147483647, -2147483646,
123456789012,
      'PS comment field is 199 long no E');
select * from partsupp
      where ps_partkey = 2147483647
      and ps_suppkey = 2147483647;
delete from partsupp
      where ps_partkey = 2147483647
      and ps_suppkey = 2147483647;
rem
================================
================================
insert into customer
      (c_custkey, c_name, c_address,
c_nationkey,
      c_phone, c_acctbal, c_mktsegment,
c_comment)
      values
```

```
        (2147483647, 'Customer Name goes to 25E',
        'Customer Address goes here..3.....4E',
        2147483647, 'This is phone E', 123456789012,
        'ZMark segE', 'Customer comments fiels is 117 long no E');
select * from customer
        where c_custkey = 2147483647;
delete from customer
        where c_custkey = 2147483647;
rem
================================
================================
insert into orders
        (o_orderkey, o_custkey, o_orderstatus, o_totalprice,
        o_orderdate, o_orderpriority, o_clerk, o_shippriority,
        o_comment)
        values
        (2147483647, 2147483647, 'X', 123456789012,
        TO_DATE('2005-12-30','YYYY-MM-DD'),
        'Order Priorty5E', 'Fixed text 15E', -2147483646,
        'Order comments field is 79 no E');
select * from orders
        where o_orderkey = 2147483647
        and  o_custkey = 2147483647;
delete from orders
        where o_orderkey = 2147483647
        and  o_custkey = 2147483647;
rem
================================
================================
insert into lineitem
        (l_orderkey, l_partkey, l_suppkey, l_linenumber,
        l_quantity, l_extendedprice, l_discount, l_tax,
        l_returnflag, l_linestatus, l_shipdate, l_commitdate,
        l_receiptdate, l_shipinstruct, l_shipmode, l_comment)
        values
        (2147483647,
        2147483647,
        2147483647,
        -2147483646,
        -123456789012,
        -123456789012,
        -123456789012,
        -123456789012,
        'Q',
        'R',
        TO_DATE('2005-12-30','YYYY-MM-DD'),
        TO_DATE('2005-12-30','YYYY-MM-DD'),
        TO_DATE('2005-12-30','YYYY-MM-DD'),
        'Ship by camel .....5E',
        'Ship ASAPE',
        'Is this really what you wanted? 44 long....E');
select * from lineitem
        where l_orderkey = 2147483647
        and l_partkey = 2147483647
        and l_suppkey = 2147483647
        and l_linenumber = -2147483646;
delete from lineitem
        where l_orderkey = 2147483647
        and l_partkey = 2147483647
        and l_suppkey = 2147483647
        and l_linenumber = -2147483646;
rem
================================
================================
insert into nation
        (n_nationkey, n_name, n_regionkey, n_comment)
        values
        (2147483647,
        'Ze Republic d MakebelievE',
        2147483647,
        'A nation comment for field size 152 no
```

```
E');
select * from nation
        where n_nationkey = 2147483647
        and n_regionkey = 2147483647;
delete from nation
        where n_nationkey = 2147483647
        and n_regionkey = 2147483647;
rem
================================
================================
insert into region
        (r_regionkey, r_name, r_comment)
        values
        (2147483647,
        'Ze ends of the earth....E',
        'A reasonable comment would go herE');
select * from region
        where r_regionkey = 2147483647;
delete from region
        where r_regionkey = 2147483647;
rem
================================
================================

REM get timestamp
select to_char (sysdate, 'HH24:MI:SS')
timestamp from dual;
rem
================================
================================
rem Done
rem
================================
================================
spool off;
exit;
```

F.3 gen_seed.sh

```
#!/bin/ksh

SEED_FILE=$1

#Generate the seed
echo "Setting the random number seed"
PSEED=`date +%m:%d:%H:%M:%S | sed -e
's/://g'`
echo "Using ${PSEED} as seed0"
echo ${PSEED} > $SEED_FILE
echo "Done setting the random number seed"
```

F.4 gtime.c

```
#ifdef RCSID
static char *RCSid =
    "$Header: gettime.c 15-jul-99.14:27:44
mpoess Exp $ ";
#endif /* RCSID */

/* Copyright (c) Oracle Corporation 1999. All
Rights Reserved. */

/*

    NAME
      gettime.c

    DESCRIPTION
     get wall clock time.
      get cpu time.

    FUNCTIONS
      get wall clock time.
      get cpu time.

    NOTES
     Both routines return time in seconds as a
double.
    MODIFIED    (MM/DD/YY)
    mpoess      07/15/99 - Creation
    mpoess      07/15/99 - Creation

*/

/*
**  Options:
```

```
** TIME_W_TIMES: implement gettime() with times().
** TIME_W_GETTIME: implement gettime() with gettimeofday().
** CPU_W_TIMES: implement getcpu() with times().
** CPU_W_GETRU: implement getcpu() with getrusage().
** GETRU_STATS: collect getrusage statistics
** GET_P_STATS: collect get_process_stats statistics
*/

#define SUN_OS5

#if defined(SUN_OS5)
#define TIME_W_GETTIME
#define CPU_W_TIMES
#undef GETRU_STATS
#undef CPU_W_GETRU
#endif /* SUN_OS5 */

#if defined(sequent) || defined(SEQ_PSX)
# define GET_P_STATS
#endif /* sequent */

#if defined(aix) || defined(AIXRIOS)
# define TIME_W_GETTIME
# define CPU_W_TIMES
# define GETRU_STATS
#endif /* AIXRIOS */

#if defined(a_osf) || defined(A_OSF)
# define TIME_W_GETTIME
# define CPU_W_GETRU
# define GETRU_STATS
#endif /* AIXRIOS */

#if defined(HPUX) || defined(XENIX_386) || defined(SYSV_386) || defined(ATT_3B)
# define TIME_W_TIMES
# define CPU_W_TIMES
#endif /* HPUX || XENIX_386 || SYSV_386 */

#if !defined(TIME_W_GETTIME) && !defined(TIME_W_TIMES)
# define TIME_W_TIMES
#endif

#if !defined(CPU_W_GETRU) && !defined(CPU_W_TIMES)
# define CPU_W_TIMES
#endif

#ifdef GET_P_STATS
# ifdef GETRU_STATS
#  undef GETRU_STATS
# endif
#endif

#if defined(TIME_W_GETTIME) || defined(CPU_W_GETRU) || defined(GETRU_STATS)
# include <sys/time.h>
#endif /* TIME_W_GETTIME || CPU_W_GETRU || GETRU_STATS */

#if defined(CPU_W_GETRU) || defined(GETRU_STATS)
# include <sys/resource.h>
#endif /* CPU_W_GETRU || GETRU_STATS */

#if defined(TIME_W_TIMES) || defined (CPU_W_TIMES)
# include <sys/types.h>
# include <sys/times.h>
# include <sys/param.h> /* most systems define HZ here */
#endif /* TIME_W_TIMES or CPU_W_TIMES */

#ifdef GET_P_STATS
# include <sys/types.h>
# include <sys/procstats.h>
#endif /* GET_P_STATS */
```

```
# include <stdio.h>

#ifdef GETRU_STATS
struct rusage selfru;
struct rusage kidsru;
#endif /* GETRU_STATS */

#ifdef GET_P_STATS
struct process_stats selfru;
struct process_stats kidsru;
#endif /* GET_P_STATS */



double gettime ()

{

#ifdef TIME_W_GETTIME
   struct timeval tv;

   (void) gettimeofday (&tv, (struct timezone
*) 0);
    return ((double) tv.tv_sec + (1.0e-6 *
(double) tv.tv_usec));
#endif /* TIME_W_GETTIME */

#ifdef TIME_W_TIMES
   struct tms buf;

  return ((double) times (&buf) / HZ);
#endif /* TIME_W_TIMES */

}



double getcpu ()

{

#ifdef CPU_W_TIMES
   struct tms buf;

   (void) times (&buf);
   return (((double) buf.tms_utime + (double)
buf.tms_stime) / HZ);
#endif /* CPU_W_TIMES */

#ifdef CPU_W_GETRU
   struct rusage ru;
   double usecs;

   (void) getrusage (0, &ru);
    usecs = 1.0e-6 * (double)
(ru.ru_utime.tv_usec + ru.ru_stime.tv_usec);
    return ((double) (ru.ru_utime.tv_sec +
ru.ru_stime.tv_sec) + usecs);
#endif /* CPU_W_GETRU */

}



getru (fp, kids, config, runname, proc_no)

FILE *fp;
int kids;
char *config;
char *runname;
int proc_no;

{

#ifdef GETRU_STATS
   struct rusage ru;

    fprintf (fp, "%-10.10s %-10.10s %10d
%10d ", config,runname, proc_no, kids);
   getrusage (kids ? RUSAGE_CHILDREN :
RUSAGE_SELF, &ru);
    print_ru (fp, &ru);
   fprintf (fp, "\n");
#endif /* GETRU_STATS */

#ifdef GET_P_STATS
   timeval_t tv;
```

```
   struct process_stats ru;

    fprintf (fp, "%-10.10s %-10.10s %10d %10d ", config,runname, proc_no, kids);
   if (kids)
       get_process_stats (&tv, PS_SELF, (struct process_stats *) 0, &ru);
    else
       get_process_stats (&tv, PS_SELF, &ru, (struct process_stats *) 0);
   print_ru (fp, &ru);
   fprintf (fp, "\n");
#endif /* GET_P_STATS */

}



getru1 (kids)

int kids;

{

#ifdef GETRU_STATS
   if (kids) {
      memset (&kidsru, 0, sizeof (kidsru));
       getrusage (RUSAGE_CHILDREN, &kidsru);
    }
   else {
      memset (&selfru, 0, sizeof (selfru));
       getrusage (RUSAGE_SELF, &selfru);
    }
#endif /* GETRU_STATS */

#ifdef GET_P_STATS
   timeval_t tv;

   if (kids) {
      memset (&kidsru, 0, sizeof (kidsru));
       get_process_stats (&tv, PS_SELF, (struct process_stats *) 0, &kidsru);
    }
   else {
      memset (&selfru, 0, sizeof (selfru));
       get_process_stats (&tv, PS_SELF, &selfru, (struct process_stats *) 0);
    }
#endif /* GET_P_STATS */

}



getru2 (fp, kids, config, runname, proc_no)

FILE *fp;
int kids;
char *config;
char *runname;
int proc_no;

{

#ifdef GETRU_STATS
   struct rusage ru;

    fprintf (fp, "%-10.10s %-10.10s %10d %10d ", config, runname, proc_no, kids);
   getrusage (kids ? RUSAGE_CHILDREN : RUSAGE_SELF, &ru);
   if (kids)
      diffru (&ru, &kidsru);
    else
      diffru (&ru, &selfru);
   print_ru (fp, &ru);
   fprintf (fp, "\n");
#endif /* GETRU_STATS */

#ifdef GET_P_STATS
   timeval_t tv;
   struct process_stats ru;

    fprintf (fp, "%-10.10s %-10.10s %10d %10d ", config, runname, proc_no, kids);
   if (kids)
       get_process_stats (&tv, PS_SELF,
```

```
(struct process_stats *) 0, &ru);
    else
        get_process_stats (&tv, PS_SELF, &ru,
(struct process_stats *) 0);
   if (kids)
       diffru (&ru, &kidsru);
    else
       diffru (&ru, &selfru);
   print_ru (fp, &ru);
   fprintf (fp, "\n");
#endif /* GET_P_STATS */

}



#ifdef GETRU_STATS

print_ru (fp, ru)

FILE *fp;
struct rusage *ru;

{

    fprintf (fp, "%10ld ", ru->ru_utime.tv_sec
* 1000 +

(ru->ru_utime.tv_usec/1000));
   fprintf (fp, "%10ld ", ru->ru_stime.tv_sec *
1000 +

(ru->ru_stime.tv_usec/1000));
   fprintf (fp, "%10ld ", ru->ru_maxrss);
   fprintf (fp, "%10ld ", ru->ru_majflt);
   fprintf (fp, "%10ld ", ru->ru_minflt);
   fprintf (fp, "%10ld ", 0);
   fprintf (fp, "%10ld ", 0);
   fprintf (fp, "%10ld ", 0);
   fprintf (fp, "%10ld ", ru->ru_nswap);
    fprintf (fp, "%10ld ", 0);
   fprintf (fp, "%10ld ", ru->ru_nvcsw);
   fprintf (fp, "%10ld ", ru->ru_nivcsw);
   fprintf (fp, "%10ld ", ru->ru_nsignals);
   fprintf (fp, "%10ld ", 0);
   fprintf (fp, "%10ld ", 0);
   fprintf (fp, "%10ld ", ru->ru_inblock);
   fprintf (fp, "%10ld ", ru->ru_oublock);
   fprintf (fp, "%10ld ", 0);
    fprintf (fp, "%10ld",   0);

}



diffru (ru2, ru)

struct rusage *ru2;
struct rusage *ru;

{

    ru2->ru_utime.tv_sec                    -=
ru->ru_utime.tv_sec;
    ru2->ru_utime.tv_usec                   -=
ru->ru_utime.tv_usec;
    ru2->ru_stime.tv_sec                    -=
ru->ru_stime.tv_sec;
    ru2->ru_stime.tv_usec                   -=
ru->ru_stime.tv_usec;
    ru2->ru_maxrss -= ru->ru_maxrss;
    ru2->ru_ixrss -= ru->ru_ixrss;
    ru2->ru_idrss -= ru->ru_idrss;
    ru2->ru_minflt -= ru->ru_minflt;
    ru2->ru_majflt -= ru->ru_majflt;
    ru2->ru_nswap -= ru->ru_nswap;
    ru2->ru_inblock -= ru->ru_inblock;
    ru2->ru_oublock -= ru->ru_oublock;
    ru2->ru_msgsnd -= ru->ru_msgsnd;
    ru2->ru_msgrcv -= ru->ru_msgrcv;
    ru2->ru_nsignals -= ru->ru_nsignals;
    ru2->ru_nvcsw -= ru->ru_nvcsw;
    ru2->ru_nivcsw -= ru->ru_nivcsw;

}

#endif /* GETRU_STATS */
```

```
#ifdef GET_P_STATS

print_ru (fp, ps)

FILE *fp;
struct process_stats *ps;

{

    fprintf (fp, "%lu ", ps->ps_utime.tv_sec *
1000 +

(ps->ps_utime.tv_usec/1000));
    fprintf (fp, "%lu ", ps->ps_stime.tv_sec *
1000 +

(ps->ps_stime.tv_usec/1000));
   fprintf (fp, "%lu ", ps->ps_maxrss);
   fprintf (fp, "%lu ", ps->ps_pagein);
   fprintf (fp, "%lu ", ps->ps_reclaim);
   fprintf (fp, "%lu ", ps->ps_zerofill);
   fprintf (fp, "%lu ", ps->ps_pffincr);
   fprintf (fp, "%lu ", ps->ps_pffdecr);
   fprintf (fp, "%lu ", ps->ps_swap);
   fprintf (fp, "%lu ", ps->ps_syscall);
   fprintf (fp, "%lu ", ps->ps_volcsw);
   fprintf (fp, "%lu ", ps->ps_involcsw);
   fprintf (fp, "%lu ", ps->ps_signal);
   fprintf (fp, "%lu ", ps->ps_lread);
   fprintf (fp, "%lu ", ps->ps_lwrite);
   fprintf (fp, "%lu ", ps->ps_bread);
   fprintf (fp, "%lu ", ps->ps_bwrite);
    fprintf (fp, "%lu ", ps->ps_phread);
   fprintf (fp, "%lu",   ps->ps_phwrite);

}



diffru (ru2, ru)

struct process_stats *ru2;
struct process_stats *ru;

{

    ru2->ps_utime.tv_sec                    -=
ru->ps_utime.tv_sec;
    ru2->ps_utime.tv_usec                   -=
ru->ps_utime.tv_usec;
    ru2->ps_stime.tv_sec                    -=
ru->ps_stime.tv_sec;
    ru2->ps_stime.tv_usec                   -=
ru->ps_stime.tv_usec;
    ru2->ps_maxrss -= ru->ps_maxrss;
    ru2->ps_pagein -= ru->ps_pagein;
    ru2->ps_reclaim -= ru->ps_reclaim;
    ru2->ps_zerofill -= ru->ps_zerofill;
    ru2->ps_pffincr -= ru->ps_pffincr;
    ru2->ps_pffdecr -= ru->ps_pffdecr;
    ru2->ps_swap -= ru->ps_swap;
    ru2->ps_syscall -= ru->ps_syscall;
    ru2->ps_volcsw -= ru->ps_volcsw;
    ru2->ps_involcsw -= ru->ps_involcsw;
    ru2->ps_signal -= ru->ps_signal;
    ru2->ps_lread -= ru->ps_lread;
    ru2->ps_lwrite -= ru->ps_lwrite;
    ru2->ps_bread -= ru->ps_bread;
    ru2->ps_bwrite -= ru->ps_bwrite;
    ru2->ps_phread -= ru->ps_phread;
    ru2->ps_phwrite -= ru->ps_phwrite;


}

#endif /* GET_P_STATS */
```

F.5 qexecpl.c

```
#ifdef RCSID
static char *RCSid =
    "$Header:  qexecpl.c  17-oct-2001.09:29:47
mpoess Exp $ ";
#endif /* RCSID */

/*  Copyright  (c)  1999,  2001,  Oracle
```

```
Corporation.  All rights reserved.  */

/*

    NAME
      qexecpl.c - <one-line expansion of the
name>

    DESCRIPTION
      SQL Execution Engine, Oracle v8,
OCI version

   PRIVATE FUNCTION(S)
     <list of static functions defined in .c file
- with one-line descriptions>

    MODIFIED     (MM/DD/YY)
       mpoess      10/17/01 - add serialization
level in SQLinit
       mpoess          02/22/01 - add linux
changes
       mpoess      08/05/99 - make compile
       mpoess          11/13/98 - fix pddl
statement
       pswong      02/19/97 - migrating to
version 8
       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 <errno.h>
#include <math.h>
#include <string.h>
#include <sys/types.h>
#include <time.h>
#include <stdlib.h>

#include "qexecpl.h"

/* Function Prototypes */

extern double gettime();

/* function prototypes from gen.c */

int get_statement();

/* Declare error handling functions */

void sql_error();

/* 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 */
char *passwd;

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 qn[3];                  /* Number of the
query being executed  */
char qnp[3];                 /* Number of the
previous query executed  */
char cmnt[5000];                /* Buffer to
save the comment.           */
#ifdef LINUX
FILE *qtemp;         /* fd for query template
*/
FILE *logfile;     /* log and report files
*/
FILE *rep;
#else
FILE *qtemp = stdin;         /* fd for query
template                    */
FILE *logfile = stdout;    /* log and report
files                       */
FILE *rep = stdout;
#endif
void *defbuf;                /* Buffer pointer
for ODEFIN              */
int deflen = 0;              /* Size of data type
for ODEFIN          */
int deftype = 1;             /* Oracle type
number for ODEFIN        */

int pfmem = PFMEMSIZE;     /* Memory to
prefetch rows               */

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

/* OCI handles */

OCIEnv *tpcenv = NULL;
OCIServer *tpcsrv = NULL;
OCIError *errhp = NULL;
OCISvcCtx *tpcsvc = NULL;
OCISession *tpcusr = NULL;
OCIStmt *curq = NULL;
OCIStmt *cur_dml = NULL;
OCIStmt *cur_ddl = NULL;
OCIParam *tpcpar = NULL;

sword status = OCI_SUCCESS;  /* OCI
return value */

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

void usage() {

   fprintf(stderr,"\nUsage:                qexec
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);
}



/* type: 0 if environment handle is passed, 1 if error handle is passwd */

void sql_error(errhp,status,type)
      OCIError *errhp;
      sword status;
      sword type;
{
  char msg[2048];
  ub4 errcode;
  ub4 msglen;
  int i,j;

  switch(status) {
  case OCI_SUCCESS_WITH_INFO:
     fprintf(stderr, "Error: Statement returned with info.\n");
     if (type)
       (void) OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)msg,
                  2048,OCI_HTYPE_ERROR);
     else
       (void) OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)msg,
                  2048,OCI_HTYPE_ENV);
     fprintf(stderr,"%s\n",msg);
     break;
  case OCI_ERROR:
    fprintf(stderr, "Error: OCI call error.\n");
    if (type)
       (void) OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)msg,
                  2048,OCI_HTYPE_ERROR);
     else
       (void) OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)msg,
                  2048,OCI_HTYPE_ENV);
     fprintf(stderr,"%s\n",msg);
     break;
  case OCI_INVALID_HANDLE:
    fprintf(stderr, "Error: Invalid Handle.\n");
    if (type)
       (void) OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)msg,
                  2048,OCI_HTYPE_ERROR);
     else
       (void) OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)msg,
                  2048,OCI_HTYPE_ENV);
     fprintf(stderr,"%s\n",msg);
     break;
   }

  /* Rollback just in case */

  (void) OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);

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

```
   SQLexit();

   exit(1);
}

#ifdef LINUX
int main(argc,argv)
#else
void main(argc,argv)
#endif
        int argc;
        char *argv[];
{

  int i,pos,pos2;
    int retcode;            /* Return code for
get_statement */
#ifdef LINUX
   logfile=fopen("/dev/stdout","w");
   qtemp=fopen("/dev/stdin","rw");
   rep=fopen("/dev/stdout","w");
#endif
  /* 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));
  fprintf(rep, "Begin Executing this Stream 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;
       } */
       if (stmt[3]== '@') {
           pos=4;
           strcpy(qnp,qn);
          while (stmt[pos] != ')') {
             pos++;
           }
           pos2=0;
           pos++;
          while (stmt[pos] != '.') {
              /*printf ("qn %d %c
\n",pos2,stmt[pos]);*/
              qn[pos2]=stmt[pos];
              pos2++;
              pos++;
           }
          qn[pos2] = 0;
            /* printf("found a new query:
%s\n",qn); */
        }
       /* save the comment so that we can
print it out later on  */
       strcat(cmnt, 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();

       stmt_cnt++;
       qry_cnt++;
       fflush(rep);
       fflush(logfile);
       /*
       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));
      fprintf(rep, "Query %s: Execution Time:
%.2f started %.2f ended %.2f\n",
                  qn,(s_tr_end -
s_tr_start)s_tr_start,s_tr_end);
       fflush(rep);
       fflush(logfile);*/
       break;

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

```
  /* Get Timing for the last query */

  tr_end = gettime();

   fprintf(logfile,"Query Processed in %.2f seconds.\n\n",(tr_end - s_tr_start));

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

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

  /* fprintf(rep, "Query %s : Execution time %.2f\n", qn,(tr_end - s_tr_start));*/
    fprintf(rep, "Query %s: Execution Time: %.2f started %.2f ended %.2f\n",
                                qn,(tr_end - s_tr_start),s_tr_start,tr_end);

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

   fprintf(rep,"\nEnded Executing this Stream at %s\n", ctime(&tim));
   fprintf(rep,"\nStream Started at %.2f\n", tr_start);
   fprintf(rep,"Stream Ended at %.2f\n", tr_end);
   fprintf(rep,"Stream Processed in %.2f seconds\n\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));
       dlist[i]->defhdl = NULL;
/*
OCIhalloc(curq,&(dlist[i]->defhdl),OCI_HTYPE_DEFINE); */
      }
      else
        dlist[i] = NULL;
   }
```

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

   (void) OCIInitialize(OCI_DEFAULT,(dvoid *)0,0,0,0);

   if((status=OCIEnvInit((OCIEnv **)&tpcenv,OCI_DEFAULT,0,(dvoid **)0)) !=
        OCI_SUCCESS)
     sql_error(tpcenv, status, 0);


OCIhalloc(tpcenv,&errhp,OCI_HTYPE_ERROR);

OCIhalloc(tpcenv,&curq,OCI_HTYPE_STMT);

OCIhalloc(tpcenv,&cur_dml,OCI_HTYPE_STMT);

OCIhalloc(tpcenv,&cur_ddl,OCI_HTYPE_STMT);

OCIhalloc(tpcenv,&tpcsvc,OCI_HTYPE_SVCCTX);

OCIhalloc(tpcenv,&tpcsrv,OCI_HTYPE_SERVER);

OCIhalloc(tpcenv,&tpcusr,OCI_HTYPE_SESSION);

  /* get username and password */

   passwd = strchr(logname, '/');
   *passwd = '\0';
   passwd++;

   if          ((status          =
OCIServerAttach(tpcsrv,errhp,(text *)0,0,OCI_DEFAULT)) != OCI_SUCCESS)
      sql_error(errhp,status,1);


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcsrv,0,OCI_ATTR_SERVER,errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,logname,strlen(logname),OCI_ATTR_USERNAME,
          errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI_ATTR_PASSWORD,
          errhp);

  if ((status = OCISessionBegin(tpcsvc, errhp, tpcusr, OCI_CRED_RDBMS,
                    OCI_DEFAULT ))       != OCI_SUCCESS)
      sql_error(errhp,status,1);


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcusr,0,OCI_ATTR_SESSION,errhp);

/*
   if          ((status=OCILogon((OCIEnv *)tpcenv,(OCIError       *)errhp,(OCISvcCtx *)tpcsvc,
              (text *)logname, strlen(logname), (text *)passwd,
              strlen(passwd), (text *) 0, 0)) != OCI_SUCCESS)
     sql_error(errhp, status, 1);
*/
   printf("\nConnected to ORACLE as user: %s\n\n", logname);

}
```

```
/* 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;
  ub2   stmttyp   =   OCI_STMT_SELECT;
/* default is a SELECT statement */

 /* 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.  */

  if (qry_cnt) {
      time(&tim);
     s_tr_end = gettime();
      fprintf(logfile,"Query Processed in %.2f seconds.\n\n",
       (s_tr_end - s_tr_start));

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

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

       /*fprintf(rep, "Query %s : Execution time %.2f\n", qnp,(s_tr_end - s_tr_start));*/
      fprintf(rep, "Query %s: Execution Time: %.2f started %.2f ended %.2f\n",
                        qnp,(s_tr_end - s_tr_start),s_tr_start,s_tr_end);

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

      /* Fix for Q15 */
       strcpy(qnp,qn);

       fflush(logfile);
       fflush(rep);
  }
  else
       tr_start = gettime();

  s_tr_start = gettime();

  /* prepare the statement */

   if ((status = OCIStmtPrepare(curq, errhp, (text*) stmt, (ub4) strlen(stmt),
                        OCI_NTV_SYNTAX, OCI_DEFAULT)) != OCI_SUCCESS)
      sql_error(errhp,status,1);

  /* Prints the query text and comment to the logfile */

  fprintf(logfile, "\n%s\n", cmnt);
  cmnt[0]=0;
  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          */


OCIaget(curq,OCI_HTYPE_STMT,&stmttyp, NULL,OCI_ATTR_STMT_TYPE,errhp);

  if (stmttyp != OCI_STMT_SELECT) {
     OCIsexec(tpcsvc,curq,errhp,1);
     return;
  }
```

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

   /* first let's execute it to get the select-list
definition */

   OCIaset(curq, OCI_HTYPE_STMT,
&pfmem, 0,
OCI_ATTR_PREFETCH_MEMORY, errhp);

   OCIsexec(tpcsvc,curq,errhp,0);

   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 */


OCIaget(curq,OCI_HTYPE_STMT,&rows_ret
,NULL,OCI_ATTR_ROW_COUNT,errhp);

   /* 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;

  OCILogoff(tpcsvc,errhp);
  OCIhfree(tpcenv,OCI_HTYPE_STMT);
  OCIhfree(tpcsvc,OCI_HTYPE_SVCCTX);
  OCIhfree(tpcsrv,OCI_HTYPE_SERVER);
  OCIhfree(tpcusr,OCI_HTYPE_SESSION);

 /* free all memory */

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

 /* 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;
 int retflag = 0;

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

```
    slist[i].buflen                    =
MAX_COLNAME_SIZE;

    if                  (OCIParamGet(curq,
OCI_HTYPE_STMT,  errhp,  (dvoid  **)
&tpcpar,
                        POS(i))          !=
OCI_SUCCESS)
        break;


   /* dsize and nullok fields of dlist not used
*/

    OCIaget(tpcpar,  OCI_DTYPE_PARAM,
&(slist[i].dbsize),
           NULL,  OCI_ATTR_DATA_SIZE,
errhp);
    OCIaget(tpcpar,  OCI_DTYPE_PARAM,
&(slist[i].dbtype),
           NULL,  OCI_ATTR_DATA_TYPE,
errhp);
    OCIaget(tpcpar,  OCI_DTYPE_PARAM,
&(slist[i].buf),
           &(slist[i].buflen),
OCI_ATTR_NAME, errhp);
    OCIaget(tpcpar,  OCI_DTYPE_PARAM,
&(slist[i].precision),
           NULL,  OCI_ATTR_PRECISION,
errhp);
    OCIaget(tpcpar,  OCI_DTYPE_PARAM,
&(slist[i].scale),
          NULL, OCI_ATTR_SCALE, errhp);

   /* 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));
       dlist[i]->defhdl = NULL;
    }

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

    switch (slist[i].dbtype) {

    case OCI_TYPECODE_NUMBER:

       /* 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 = OCI_TYPECODE_DOUBLE;
      slist[i].dbtype                    =
OCI_TYPECODE_DOUBLE;
       } else {
      defbuf = (int *) dlist[i]->ibuf;
      deflen = INT;
      deftype = OCI_TYPECODE_INTEGER;
      slist[i].dbtype                    =
OCI_TYPECODE_INTEGER;
       }
#else
      defbuf = (double *) dlist[i]->fbuf;
       deflen = FLT;
       deftype = OCI_TYPECODE_FLOAT;
        slist[i].dbtype                   =
OCI_TYPECODE_FLOAT;
#endif /* HAVE_SCALE */

        break;
```

```
      default:

       /* default is character string */

       defbuf = (char **) dlist[i]->sbuf;
       deflen = MAX_STR_LEN;
       deftype = SQLT_STR;
/*       deftype = OCI_TYPECODE_CHAR;
*/
        break;
      }

    /* Define the column */

     if
((status=OCIDefineByPos(curq,&(dlist[i]->def
hdl),errhp,POS(i),

defbuf,deflen,deftype,NULL,

dlist[i]->rlen,NULL,OCI_DEFAULT))!=OCI_
SUCCESS)
        sql_error(errhp,status,1);
   }
  return i;
}



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

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

  int i,j;
  int ntf;
  int num_so_far;
  sword stats = OCI_SUCCESS;

  /* 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)) {

      stats = OCIStmtFetch(curq, errhp,
MAX_ARRAY,          OCI_FETCH_NEXT,
OCI_DEFAULT);


OCIaget(curq,OCI_HTYPE_STMT,&rows_ret
,NULL,OCI_ATTR_ROW_COUNT,errhp);

      print_rows(num,rows_ret);

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

    if (stats != OCI_NO_DATA) {
       if (num_to_fetch == -1) {
      while          ((stats          =
OCIStmtFetch(curq,errhp,MAX_ARRAY,OCI
_FETCH_NEXT,
                          OCI_DEFAULT))
== OCI_SUCCESS) {

OCIaget(curq,OCI_HTYPE_STMT,&num_so_
far,NULL,

OCI_ATTR_ROW_COUNT,errhp);

print_rows(num,(num_so_far-rows_ret));
         rows_ret = num_so_far;
       }
       /* Print the final rows */
       OCIaget(curq,OCI_HTYPE_STMT,&nu
```

```
m_so_far,NULL,

OCI_ATTR_ROW_COUNT,errhp);
        print_rows(num,(num_so_far-rows_ret));
        rows_ret = num_so_far;
         } else {
        ntf -= MAX_ARRAY;

        while ((stats = OCIStmtFetch(curq,errhp,

((ntf>MAX_ARRAY) ? MAX_ARRAY:ntf),

OCI_FETCH_NEXT, OCI_DEFAULT)) ==
                    OCI_SUCCESS)  {
          ntf  -= MAX_ARRAY;

OCIaget(curq,OCI_HTYPE_STMT,&num_so_
far,NULL,

OCI_ATTR_ROW_COUNT,errhp);

print_rows(num,(num_so_far-rows_ret));
          rows_ret = num_so_far;
          if (ntf <= 0) break;
        }
        OCIaget(curq,OCI_HTYPE_STMT,&nu
m_so_far,NULL,

OCI_ATTR_ROW_COUNT,errhp);
        print_rows(num,(num_so_far-rows_ret));
        rows_ret = num_so_far;
          }
       }
   } else {
       OCIStmtFetch(curq,        errhp,        ntf,
OCI_FETCH_NEXT, OCI_DEFAULT);

OCIaget(curq,OCI_HTYPE_STMT,&rows_ret
,NULL,OCI_ATTR_ROW_COUNT,errhp);
       print_rows(num,rows_ret);
   }

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

}


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, diff;
  char colname[MAX_COLNAME_SIZE];
   int len = 0;               /* Running column length */
  int cwid = 0;

  fprintf(logfile, "\n");

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

    /* extract the column name */

      strncpy((char *)colname, (char *)slist[i].buf, slist[i].buflen);
      colname[slist[i].buflen] = '\0';

    /* 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, colname);
#endif /* FORMAT1 */
   }

   fprintf(logfile,"\n");
}


void print_rows(ncol, nrow)
```

```
        int ncol;
        int nrow;
{

  int i,j;
  int len;
  int diff;
  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 = ' ';
}
```

F.6 qexecpl.h

```
/*
 * $Header: qexecpl.h 13-nov-2001.17:52:35
mpoess Exp $
 */

/* Copyright (c) 1999, 2001, Oracle
Corporation. All rights reserved. */

/* NOTE: See 'header_template.doc' in the
'doc' dve under the 'forms'
        directory for the header file template
that includes instructions.
*/

/*
   NAME
      qexecpl.h
```

```
   DESCRIPTION
      SQL statement execution front-end
header file.

   PUBLIC FUNCTION(S)
      <list of external functions
declared/defined - with one-line descriptions>

   PRIVATE FUNCTION(S)
     <list of static functions defined in .c file
- with one-line descriptions>

   EXAMPLES

   NOTES
      <other useful comments, qualifications,
etc.>

   MODIFIED    (MM/DD/YY)
   mpoess       11/13/01  - change DOP to
84 for DML and DDL
     mpoess          02/22/01  - add linux
changes
   mpoess       08/05/99  - make compile
   mpoess       07/15/99  - Creation
   mpoess       07/15/99  - Creation

*/

/*
# ifndef S_ORACLE
#   include <s.h>
# endif
*/
#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 <oratypes.h>

#ifndef OCIDFN
#include <ocidfn.h>
#endif /* OCIDFN */

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

/* 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 */
#ifndef LINUX
#define MAX(x,y) ((x >= y) ? x : y)
#define MIN(x,y) ((x <= y) ? x : y)
#endif
/* 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 32      /* 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
{
  ub2 dbsize;
  ub4 buflen;
/*  sb2 dsize;  */
  sb4 scale;
/*  sb2 nullok; */
  OCITypeCode dbtype;
/*  text buf[MAX_COLNAME_SIZE]; */
  text *buf;
  ub1 precision;
} sltype;


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

#define MAX_ARRAY 50             /* Maximum array size for array fetch */
#define PFMEMSIZE 65536      /* Memory size of prefetch buffer     */

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

#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      */
  OCIDefine *defhdl;
} 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 OCIhalloc(envh,hndl,htyp) \
    if((status=OCIHandleAlloc((dvoid *)envh,(dvoid **)hndl,htyp,0,(dvoid **)0))!=OCI_SUCCESS) \
        sql_error(envh,status,0); \
    else \
        DISCARD 0

#define OCIhfree(hndl,htyp) \
    if((status=OCIHandleFree((dvoid *)hndl,htyp)) == OCI_SUCCESS) \
        fprintf(stderr, "Error freeing handle of type %d\n", htyp)

#define OCIaget(hndl,htyp,attp,size,atyp,errh) \
    if((status=OCIAttrGet((dvoid *)hndl,htyp,(dvoid *)attp,(dvoid *)size,atyp,errh)) != OCI_SUCCESS) \
        sql_error(errh,status,1); \
    else \
        DISCARD 0

#define OCIaset(hndl,htyp,attp,size,atyp,errh) \
    if((status=OCIAttrSet((dvoid *)hndl,htyp,(dvoid *)attp,size,atyp,errh)) != OCI_SUCCESS) \
        sql_error(errh,status,1); \
    else \
        DISCARD 0

#define OCIsexec(svch,stmh,errh,iter) \

if((status=OCIStmtExecute(svch,stmh,errh,iter,0,NULL,NULL,OCI_DEFAULT)) != OCI_SUCCESS) \
        sql_error(errh,status,1); \
    else \
        DISCARD 0



#define ISOTXT "alter session set isolation_level = serializable"
#define PDMLTXT "alter session force parallel dml parallel (degree 84)"
#define PDDLTXT "alter session force parallel ddl parallel (degree 84)"

#endif /* QSTREAMPL_H */
```

F.7 runTPCHall.beforeload

```
#!/bin/ksh
. $KIT_DIR/env

ECHO=echo

sqlplus=$ORACLE_HOME/bin/sqlplus
GTIME=${KIT_DIR}/utils/gtime

RUN_ID_FILE=${KIT_DIR}/audit/r_id

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

OUT_DIR=${KIT_DIR}/audit/tests/${RUN_ID}
if [ ! -d $OUT_DIR ]
then
  mkdir $OUT_DIR
fi

SCRIPT_LOG_FILE=${OUT_DIR}/main.out
RDB_TABLES=${OUT_DIR}/rdbtablest
FIRST_TEN=${OUT_DIR}/firstten
LD1DBCRE=${OUT_DIR}/Ld1dbcre
LD2SCTSO=${OUT_DIR}/Ld2sctso
LD3DAPOP=${OUT_DIR}/Ld3dapop
LD4IXCRE=${OUT_DIR}/Ld4ixcre
LD5ANLYZ=${OUT_DIR}/Ld5anlyz
DAT_FILE=${KIT_DIR}/audit/1TB_final.dat

echo Start TPC-H Benchmark SEQUENCE NUMBER: $RUN_ID > $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE
echo "Starting a new Oracle log file: $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log" >> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE

mv $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log.preAudit.$RUN_ID
touch $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log

echo "Start: load database `date`" >> $SCRIPT_LOG_FILE
bumpx.pl -s -o ${DAT_FILE} -p dbcre > $LD1DBCRE
bumpx.pl -s -o ${DAT_FILE} -p sctso > $LD2SCTSO
tshut
2start
STIME=`$GTIME`
echo "Start: timed load portion `date`" >> $SCRIPT_LOG_FILE
bumpx.pl -s -o ${DAT_FILE} -p dapop > $LD3DAPOP
bumpx.pl -s -o ${DAT_FILE} -p ixcre > $LD4IXCRE
bumpx.pl -s -o ${DAT_FILE} -p anlyz > $LD5ANLYZ
2shut
2start
ckpnt.sh
echo "End: timed load portion `date`" >> $SCRIPT_LOG_FILE
exit
```

F.8 runTPCHall.afterload

```
#!/bin/ksh
. $KIT_DIR/env

ECHO=echo
```

```
sqlplus=$ORACLE_HOME/bin/sqlplus
GTIME=${KIT_DIR}/utils/gtime

RUN_ID_FILE=${KIT_DIR}/audit/r_id

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

OUT_DIR=${KIT_DIR}/audit/tests/${RUN_ID}
if [ ! -d $OUT_DIR ]
then
  mkdir $OUT_DIR
fi

SCRIPT_LOG_FILE=${OUT_DIR}/main.out
RDB_TABLES=${OUT_DIR}/rdbtablest
FIRST_TEN=${OUT_DIR}/firstten
LD1DBCRE=${OUT_DIR}/Ld1dbcre
LD2SCTSO=${OUT_DIR}/Ld2sctso
LD3DAPOP=${OUT_DIR}/Ld3dapop
LD4IXCRE=${OUT_DIR}/Ld4ixcre
LD5ANLYZ=${OUT_DIR}/Ld5anlyz
DAT_FILE=${KIT_DIR}/audit/1TB_final.dat

#echo Start TPC-H Benchmark SEQUENCE NUMBER: $RUN_ID > $SCRIPT_LOG_FILE
#echo >> $SCRIPT_LOG_FILE
#echo "Starting a new Oracle log file: $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log" >> $SCRIPT_LOG_FILE
#echo >> $SCRIPT_LOG_FILE
#
#mv $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log.preAudit.$RUN_ID
#touch $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log
#
#echo "Start: load database `date`" >> $SCRIPT_LOG_FILE
#bumpx.pl -s -o ${DAT_FILE} -p dbcre > $LD1DBCRE
#bumpx.pl -s -o ${DAT_FILE} -p sctso > $LD2SCTSO
#tshut
#2start
#STIME=`$GTIME`
#echo "Start: timed load portion `date`" >> $SCRIPT_LOG_FILE
#bumpx.pl -s -o ${DAT_FILE} -p dapop > $LD3DAPOP
#bumpx.pl -s -o ${DAT_FILE} -p ixcre > $LD4IXCRE
#bumpx.pl -s -o ${DAT_FILE} -p anlyz > $LD5ANLYZ
#2shut
#2start
#ckpnt.sh
#echo "End: timed load portion `date`" >> $SCRIPT_LOG_FILE

2start
ckpnt.sh
$KIT_DIR/audit/gen_seed.sh
$KIT_DIR/audit/seed
echo Generated seed: `cat $KIT_DIR/audit/seed` >> $SCRIPT_LOG_FILE

echo "Start: dbtables.sql and count.sql" >> $SCRIPT_LOG_FILE
$sqlplus ${DATABASE_USER}@$KIT_DIR/audit/dbtables > ${RDB_TABLES} 2>&1
$sqlplus ${DATABASE_USER}@$KIT_DIR/audit/firstten > ${FIRST_TEN}
```

```
2>&1
echo "End: dbtables.sql and count.sql `date`"
>> $SCRIPT_LOG_FILE

#2shut >>  $SCRIPT_LOG_FILE
#2start >>  $SCRIPT_LOG_FILE
#ckpnt.sh
runTPCHpt  ${SCALE_FACTOR}  1
${RUN_ID}

#2shut >>  $SCRIPT_LOG_FILE
#2start >>  $SCRIPT_LOG_FILE
ckpnt.sh
runTPCHpt  ${SCALE_FACTOR}  2
${RUN_ID}

sleep 600
2shut >> $SCRIPT_LOG_FILE

cp
$ORACLE_HOME/rdbms/log/alert_${ORAC
LE_SID}.log $OUT_DIR

echo "End TPC-H Benchmark SEQUENCE
NUMBER:  $RUN_ID  `date`"  >>
$SCRIPT_LOG_FILE
```

F.9 runTPCHpt

```
#!/bin/ksh
. $KIT_DIR/env
#set -x
#ECHO=/bin/echo
SCRIPT_DIR=${KIT_DIR}/scripts
SQL_DIR=${KIT_DIR}/sql
UPD_DIR=${KIT_DIR}/update
SRC_DIR=${KIT_DIR}/utils
QRY_DIR=${KIT_DIR}/queries # this is the
location of the query template file
QGEN_DIR=${KIT_DIR}/dbgen
QGEN=${QGEN_DIR}/qgen
QEXEC=${SRC_DIR}

DSS_QUERY=${KIT_DIR}/queries
export DSS_QUERY

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

TPCD_BIN=${KIT_DIR}/audit/bin

GTIME=${SRC_DIR}/gtime
SEED_FILE=${KIT_DIR}/audit/seed

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

# The defaults

QPROG=${QEXEC}/qexec

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>
<run_number>"
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
$U2PROG."
echo "-o                  : Collect Oracle
statistics."
echo "-s                  : Collect System
statistics."
echo "-u <user/passwd>  : User/Password.
Default is tpch/tpch."
echo "-h                  : Displays this
message."
}
set -- `getopt "p:u1:u2:osu:h" "$@"` || usage

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



if [ "$#" -ne "3" ]
then
  usage
  exit 1
fi


SF=$1
PARA=$2
RUN_ID=$3

OUT_DIR=${KIT_DIR}/audit/tests/${RUN_I
D}
if [ ! -d $OUT_DIR ]
then
      mkdir $OUT_DIR
fi
                                    TPCD_LOG=${OUT_DIR}
                                    TPCD_RPT=${OUT_DIR}
                                    OUT=${OUT_DIR}

                                    let
                                    UF_SET="($PARA-1)*($NUM_STREAMS+1
                                    )+1"
                                    START_SET=1
                                    let STOP_SET=$NUM_STREAMS
                                    let
                                    START_SET_UPDATE="($PARA-1)*($NUM
                                    _STREAMS+1)+2"
                                    let
                                    STOP_SET_UPDATE="$START_SET_UPDA
                                    TE+$NUM_STREAMS-1"

                                    TPCD_LOG_FILE=${TPCD_LOG}/m${PAR
                                    A}s0
                                    TPCD_RPT_FILE=${TPCD_RPT}/m${PAR
                                    A}s0inter
                                    QRY_FILE=${TPCD_RPT}/qtemp.${PARA}
                                    s0
                                    QUERY_PARAMETER=${TPCD_LOG}/qp$
                                    {PARA}.0
                                    SCRIPT_LOG_FILE=${TPCD_LOG}/m${PA
                                    RA}timing
                                    UF1_LOG=${TPCD_LOG}/m${PARA}s0rf1
                                    UF2_LOG=${TPCD_LOG}/m${PARA}s0rf2
                                    STREAM_COUNT_LOG=${TPCD_LOG}/m
                                    ${PARA}tstrcnt

                                    echo "TPC-H Test - RUN:${PARA}
                                    SEQUENCE:${RUN_ID}  `date`" >
                                    $SCRIPT_LOG_FILE
                                    echo "TPC-H Test - RUN:${PARA}
                                    SEQUENCE:${RUN_ID}  `date`" >
                                    $TPCD_RPT_FILE
                                    echo "Generates query template file with seed:
                                    `cat $SEED_FILE` for stream 0" >>
                                    $SCRIPT_LOG_FILE
                                    echo >> $SCRIPT_LOG_FILE

                                    ${QGEN} -c -r `cat $SEED_FILE` -p 0 -s
```

```
${SF} -l $QUERY_PARAMETER > ${QRY_FILE}

START=`$GTIME`
echo "Start Power Test - RUN:${PARA} SEQUENCE:${RUN_ID} Execution Starts $START, `date`" >> $SCRIPT_LOG_FILE
echo "" >> $SCRIPT_LOG_FILE

# Execute UF1

SDATE=`date`
UF1_START=`$GTIME`
echo "Start UF1 $UF1_START, `date`" >> $SCRIPT_LOG_FILE

${ECHO} ${UPD_SPT}/runuf1.sh ${UF_SET} >> $UF1_LOG 2>&1
# Execute Query Stream

UF1_END=`${GTIME}`
E1DATE=`date`

UF1_TIME=`echo $UF1_END - $UF1_START | bc`
echo UF1: Execution Time: $UF1_TIME >> ${TPCD_RPT_FILE}
echo Start Time: $UF1_START, $SDATE >> ${TPCD_RPT_FILE}
echo End Time: $UF1_END, $E1DATE >> ${TPCD_RPT_FILE}
echo "" >> ${TPCD_RPT_FILE}

echo "End UF1 $UF1_END, ${E1DATE}" >> $SCRIPT_LOG_FILE
echo UF1: Execution Time: $UF1_TIME >> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE

echo "Start Query Part `$GTIME`, `date` " >> $SCRIPT_LOG_FILE

${QPROG} ${DATABASE_USER} q${QRY_FILE} l${TPCD_LOG_FILE} r${TPCD_RPT_FILE} > $DF 2>&1

# Execute UF2

UF2_START=`${GTIME}`
E2DATE=`date`

echo "End Query Part `$GTIME`, ${E2DATE}" >> $SCRIPT_LOG_FILE
echo "" >> $SCRIPT_LOG_FILE

echo "Start UF2 $UF2_START, `date`" >> $SCRIPT_LOG_FILE
${ECHO} ${UPD_SPT}/runuf2.sh ${UF_SET} >> $UF2_LOG 2>&1
UF2_END=`${GTIME}`
END=`${GTIME}`
EDATE=`date`

UF2_TIME=`echo $UF2_END - $UF2_START | bc`
echo UF2: Execution Time: $UF2_TIME >> ${TPCD_RPT_FILE}
echo Start Time: $UF2_START, $E2DATE >> ${TPCD_RPT_FILE}
echo End Time: $UF2_END, $EDATE >> ${TPCD_RPT_FILE}


echo "End UF2 $UF2_END, $EDATE" >> $SCRIPT_LOG_FILE
echo UF2: Execution Time: $UF2_TIME >> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE

echo "End TPC-H Power Test - RUN:${PARA} SEQUENCE:${RUN_ID}, $END, $EDATE" >> $SCRIPT_LOG_FILE
MEA_INT=`echo $END - $START | bc`
echo "Elapsed Time for TPC-H Power Test - RUN:${PARA} SEQUENCE:${RUN_ID} is $MEA_INT" >> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE
```

```
${KIT_DIR}/audit/abridge.pl
${TPCD_LOG_FILE}

i=$START_SET
PSEED=`cat $SEED_FILE`

while [ $i -le $STOP_SET ]; do

TPCD_LOG_FILE=${TPCD_LOG}/mt${RU
N_ID}_${i}.log

TPCD_RPT_FILE=${TPCD_RPT}/mt${RUN
_ID}_${i}.rpt
      QUERY_PARAMETER=${TPCD_LOG
}/qp${PARA}.${i}

QRY_FILE=${TPCD_RPT}/qtemp.${PARA}
s${i}

  PSEED=`expr $PSEED + 1`
  ${QGEN} -c -r ${PSEED} -p ${i} -s ${SF}
-l $QUERY_PARAMETER > ${QRY_FILE}

  i=`expr $i + 1`
done

TH_START_D=`date`
TH_START_T=`${GTIME}`
echo >> $SCRIPT_LOG_FILE

rm -f /tmp/th_pipe1
mknod /tmp/th_pipe1 p
rm -f /tmp/th_pipe2
mknod /tmp/th_pipe2 p
i=$START_SET

echo "Start Throughput Test - RUN:${PARA}
SEQUENCE:${RUN_ID}      $TH_START_T,
$TH_START_D"   >> $SCRIPT_LOG_FILE

# starts a script to count the streams during the
throughtput run
(scnt.sh      $PARA      $RUN_ID      >
$STREAM_COUNT_LOG &)

while [ $i -le $STOP_SET ]; do
  M_SDATE=`date`
  M_STIME=`${GTIME}`
     TPCD_LOG_FILE=${TPCD_LOG}/m${
PARA}s${i}
     TPCD_RPT_FILE=${TPCD_RPT}/m${
PARA}s${i}inter
  echo "Start Query Stream $i $M_STIME,
${M_SDATE}" >> $SCRIPT_LOG_FILE
     QRY_FILE=${TPCD_RPT}/qtemp.${PA
RA}s${i}
  ${QPROG}      ${DATABASE_USER}
q${QRY_FILE}      l${TPCD_LOG_FILE}
r${TPCD_RPT_FILE} | grep -v "Connected to
ORACLE" >> $SCRIPT_LOG_FILE &
  i=`expr $i + 1`
done

(${KIT_DIR}/audit/runTPCHus      $RUN_ID
$START_SET_UPDATE
$STOP_SET_UPDATE      ${SF}      $PARA      >>
$SCRIPT_LOG_FILE 2>&1 &)

wait
THQ_END_T=`$GTIME`
THQ_END_D=`date`
echo End all Query Streams $THQ_END_T,
$THQ_END_D >> $SCRIPT_LOG_FILE
print > /tmp/th_pipe1
read < /tmp/th_pipe2

TH_END_D=`date`
TH_END_T=`$GTIME`
echo End Update Stream ${TH_END_T},
${TH_END_D} >> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE
echo "End Throughput Test ${TH_END_T},
${TH_END_D}" >> $SCRIPT_LOG_FILE
echo Execution Time Throughput Test: `echo
${TH_END_T} - ${TH_START_T} | bc` >>
$SCRIPT_LOG_FILE

i=$START_SET
```

```
while [ $i -le $STOP_SET ]; do
      TPCD_LOG_FILE=${TPCD_LOG}/m${
PARA}s${i}
  ${KIT_DIR}/audit/abridge.pl
${TPCD_LOG_FILE}
      i=`expr $i + 1`
done
PIDS=`ps -fu oracle | grep scnt.sh | grep -v
grep | awk '{print $2}'`
kill -9 $PIDS
#calculate the metric
#analyze_streams.pl -f p -n $RUN_ID >
${TPCD_RPT}/tpch_metric.${RUN_ID}.${H
ID}.rpt
```

F.10 runTPCHus

```
#!/bin/ksh
. $KIT_DIR/env

SCRIPT_DIR=${KIT_DIR}/scripts
SQL_DIR=${KIT_DIR}/sql
UPD_DIR=${KIT_DIR}/update
UPD_SPT=${UPD_DIR}/scripts
SRC_DIR=${KIT_DIR}/utils
QRY_DIR=${KIT_DIR}/queries # this is the
location of the query template file
QGEN_DIR=${KIT_DIR}/dbgen
QGEN=${QGEN_DIR}/qgen

DSS_QUERY=${KIT_DIR}/queries
export DSS_QUERY

RUN_ID=$1
START_SET_UPDATE=$2
STOP_SET_UPDATE=$3
SF=$4
PARA=$5

OUT_DIR=${KIT_DIR}/audit/tests/${RUN_I
D}
if [ ! -d $OUT_DIR ]
then
      mkdir $OUT_DIR
fi


TPCD_RPT=$OUT_DIR
SCRIPT_LOG_FILE=${OUT_DIR}/m${PAR
A}timing
OUT=$OUT_DIR

GTIME=${SRC_DIR}/gtime
HID=1

START=`$GTIME`
echo "Start Update Stream $START, `date`" >>
$SCRIPT_LOG_FILE
echo "" >> $SCRIPT_LOG_FILE

#waiting for all the query streams to finish first
read < /tmp/th_pipe1

i=$START_SET_UPDATE
j=1
while [ $i -le $STOP_SET_UPDATE ]; do

  # Execute UF1


UF1_LOG=${OUT_DIR}/m${PA RA}s${j}rf
1

UF2_LOG=${OUT_DIR}/m${PARA}s${j}rf
2

RPT_FILE=${OUT_DIR}/m${PARA}s${j}in
ter

  SDATE=`date`
  UF1_START=`$GTIME`
     echo "Start UF1-${j} at ${UF1_START},
${SDATE}" >> ${RPT_FILE}

  ${UPD_SPT}/runuf1.sh ${i} >>
${UF1_LOG} 2>&1
  UF1_END=`${GTIME}`
```

```
      EDATE=`date`
  echo "End UF1-${j} at ${UF1_END}, ${EDATE}" >> ${RPT_FILE}
      echo UF1-${j} Execution Time: `echo ${UF1_END} - ${UF1_START} | bc` >> ${RPT_FILE}

  # Execute UF2

  SDATE=`date`
  UF2_START=`${GTIME}`
  echo "Start UF2-${j} ${UF2_START}, ${SDATE}" >> ${RPT_FILE}

  ${UPD_SPT}/runuf2.sh ${i} >> ${UF2_LOG} 2>&1
  UF2_END=`${GTIME}`
  EDATE=`date`
  echo "End UF2-${j} at $UF2_END, ${EDATE}" >> ${RPT_FILE}
      echo UF2-${j} Execution Time: `echo ${UF2_END} - ${UF2_START} | bc` >> ${RPT_FILE}

  i=`expr $i + 1`
      j=`expr $j + 1`
done

print > /tmp/th_pipe2
```

F.11 runuf1.sh

```
#!/bin/ksh
#
# $Header: runuf1.sh 25-oct-2001.15:56:04 mpoess Exp $
#
# runuf1.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation. All rights reserved.
#
#      NAME
#          runuf1.sh - <one-line expansion of the name>
#
# DESCRIPTION
#      runuf1.sh -l [<path name for reports>] -u [<uid/passwd>]
#                -p [<program>] <run_id> <scale factor> <pair number>
#                <parallelism>
# USAGE
#     To execute UF1.
#
#      NOTES
#                <other useful comments, qualifications, etc.>
#
#      MODIFIED     (MM/DD/YY)
#      mpoess       10/25/01  - change default directory for update sets
#      mpoess       10/17/01  - add support for external tables
#      mpoess       08/15/99  - Creation
#      mpoess       08/15/99  - Creation
#
#
. $KIT_DIR/env
O=${ORACLE_HOME}
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
UTILS_DIR=${KIT_DIR}/utils
LOG_DIR=${UPDATE_DIR}/log
GTIME=${UTILS_DIR}/gtime
SF=${SCALE_FACTOR}
PAR_HINT=${UPDATE_DOP_1}

LOGPATH=.
PASSWD=${DATABASE_USER}

if [ $# -lt 1 ];
then
   echo runuf1.sh setnum
    exit 1
fi
SETNUM=$1
i=1
```

```
PID=""

# perform the update function 1

START=`$GTIME`

# first create the temp tables

sqlplus /NOLOG << !

connect $PASSWD;
set timing on
set serveroutput on
set echo on

drop directory data_dir;
create directory data_dir as
'/oracle.data1/flatfiles/updates/';

drop table temp_l_et;
create table temp_l_et(
    l_orderkey          number ,
    l_partkey           number ,
    l_suppkey           number ,
    l_linenumber        number ,
    l_quantity          number ,
    l_extendedprice     number ,
    l_discount          number ,
    l_tax               number ,
    l_returnflag        char(1) ,
    l_linestatus        char(1) ,
    l_shipdate          date ,
    l_commitdate        date ,
    l_receiptdate       date ,
    l_shipinstruct      char(25) ,
    l_shipmode          char(10) ,
    l_comment           varchar(44)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                records delimited by newline
                nobadfile
                nologfile
                fields terminated by '|'
                missing field values are null
        )
        location (
        'lineitem.tbl.u${SETNUM}'
))
reject limit unlimited;

drop table temp_o_et;
create table temp_o_et(
    o_orderkey          number ,
    o_custkey           number ,
    o_orderstatus       char(1) ,
    o_totalprice        number ,
    o_orderdate         date ,
    o_orderpriority     char(15) ,
    o_clerk             char(15) ,
    o_shippriority      number ,
    o_comment           varchar(79)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                records delimited by newline
                nobadfile
                nologfile
                fields terminated by '|'
                missing field values are null
        )
        location (
        'orders.tbl.u${SETNUM}'
))
reject limit unlimited;
alter table temp_l_et parallel ${PAR_HINT};
alter table temp_o_et parallel ${PAR_HINT};
```

```
alter session force parallel dml parallel (degree
${PAR_HINT});
alter session set isolation_level = serializable;
alter session set optimizer_index_cost_adj =
10;

insert into orders (
  select
    o_orderdate          ,
    o_orderkey           ,
    o_custkey            ,
    o_orderpriority      ,
    o_shippriority       ,
    o_clerk              ,
    o_orderstatus        ,
    o_totalprice         ,
    o_comment
from temp_o_et);

insert into lineitem (
  select
    l_shipdate           ,
    l_orderkey           ,
    l_discount           ,
    l_extendedprice      ,
    l_suppkey            ,
    l_quantity           ,
    l_returnflag         ,
    l_partkey            ,
    l_linestatus         ,
    l_tax                ,
    l_commitdate         ,
    l_receiptdate        ,
    l_shipmode           ,
    l_linenumber         ,
    l_shipinstruct       ,
    l_comment
from temp_l_et);

commit;

drop table temp_l_et;
drop table temp_o_et;
exit;
!

END=`$GTIME`

# Done

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

F.12 runuf2.sh

```
#!/bin/ksh
#
# $Header: runuf2.sh 25-oct-2001.15:56:05
mpoess Exp $
#
# runuf2.sh
#
# Copyright (c) 1999, 2001, Oracle
Corporation.  All rights reserved.
#
#      NAME
#        runuf2.sh - <one-line expansion of the
name>
#
# DESCRIPTION
#      runuf2.sh [-u <uid/passwd to login>] [-p
<program>] <run_id>
#                          <scale factor> <pair
number> <parallelism>
# USAGE
#     To execute UF2.
#
#     NOTES
#                <other useful comments,
qualifications, etc.>
#
#     MODIFIED    (MM/DD/YY)
```

```
#       mpoess          10/25/01  - change default
directory for update sets
#       mpoess          10/17/01  - add support for
external tables
#       mpoess          08/15/99  - Creation
#       mpoess          08/15/99  - Creation
#
. $KIT_DIR/env
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
UTILS_DIR=${KIT_DIR}/utils
GTIME=${UTILS_DIR}/gtime
LOG_DIR=${UPDATE_DIR}/log
PAR_HINT=${UPDATE_DOP_2}
SF=${SCALE_FACTOR}
PASSWD=${DATABASE_USER}

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

SETNUM=$1

i=1
PID=""

START=`$GTIME`
# first create the temp tables

sqlplus /NOLOG << !

connect $PASSWD;
set timing  on
set serveroutput on
set echo on

drop directory data_dir;
create directory data_dir as
'/oracle.data1/flatfiles/updates/';

drop table temp_okey_et;
drop table temp_okey;

create table temp_okey_et(
      t_orderkey                number
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                  records delimited by
newline
                nobadfile
                 nologfile
               fields terminated by '|'
                    missing field values are
null
          )
          location (
           'delete.${SETNUM}'))
reject limit unlimited;

alter table temp_okey_et parallel 8;

create table temp_okey parallel 8 nologging as
select * from temp_okey_et;

create unique index i_temp_okey on
temp_okey (t_orderkey) parallel 8 nologging
compute statistics;

analyze table temp_okey estimate statistics
sample 2 percent;


alter session  force parallel dml parallel
${PAR_HINT};
alter session set isolation_level=serializable;
alter session set optimizer_index_cost_adj =
10;


delete from (select /*+ use_nl(o) */ o.rowid
from orders o, temp_okey t where
o.o_orderkey = t.t_orderkey order by 1);
```

```
delete from (select /*+ use_nl(l) */ l.rowid from lineitem l,temp_okey t where l.l_orderkey = t.t_orderkey order by 1);

commit;

drop table temp_okey;
drop table temp_okey_et;
exit;
!

END=`$GTIME`

# Done

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

F.13 gen_stream

```
#!/usr/bin/ksh
#
#             $Header:             gen_streams.sh 09-mar-00.15:54:55 mpoess Exp $
#
# gen_streams.sh
#
#  Copyright (c) Oracle Corporation 2000. All Rights Reserved.
#
#      NAME
#        gen_streams.sh
#
#      DESCRIPTION
#        This file creates scripts that simulate the mutli stream run for TPC-H/R.
#        It uses qgen to generate the sequence of queries.
#       It inserts lines to gather statistics.
#
#      NOTES
#         The resulting scripts are called by runs.
#
#      MODIFIED     (MM/DD/YY)
#      mpoess         03/09/00  - create
#      mpoess         03/09/00  - Creation
#

if [ $# != 2 -a $# != 3 ]; then
    echo "Syntax error. Usage: gen_streams <scale_factor> <nb_streams> [{query-list}]"
   echo "                  Example: gen_streams 30 2 '3,17,20'"
  exit 1
fi

set -x
QLST="-q"
if [ $# -eq 3 ]; then
  QLST="-q $3"
fi

SF=$1 # scale factor
NB_STREAMS=$2 # number of streams

COUNT=1
rm -f tpcd_stream_*.sh

while [ $COUNT -le $NB_STREAMS ]; do

  let STREAM_COUNT="COUNT-1"


STREAM_NAME="tpcd_stream_"$STREAM_COUNT".sh"

  echo "#!/bin/ksh" > $STREAM_NAME
  echo >> $STREAM_NAME
   echo             'STAT_DIR=$1'             >> $STREAM_NAME
  echo 'LABEL=$2' >> $STREAM_NAME
   echo          'NUM_STREAM=$3'          >>
```

```
$STREAM_NAME
  echo 'NB_STREAMS=$4' >> $STREAM_NAME
  echo 'STAT_DIR=$5' >> $STREAM_NAME
  echo 'START_DATE=`date`' >> $STREAM_NAME
  echo 'LOG_FILE=$STAT_DIR/times.log' >> $STREAM_NAME
  echo >> $STREAM_NAME
   echo 'svrmgrl << END_SVRMGRL' >> $STREAM_NAME
  echo >> $STREAM_NAME
  echo 'connect $FRAME_USER' >> $STREAM_NAME
  echo 'alter session force parallel dml parallel(degree 160);' >> $STREAM_NAME
  echo 'rem alter session set isolation_level=serializable;' >> $STREAM_NAME
  echo >> $STREAM_NAME

  qgen -s $SF -p $COUNT $QLST -m 1 >> $STREAM_NAME

  echo >> $STREAM_NAME
  echo 'END_SVRMGRL' >> $STREAM_NAME
  echo >> $STREAM_NAME

  let COUNT="COUNT+1"
done
```

F.14 2start

```
#!/bin/ksh

. $FRAME_PATH/env

sqlplus /NOLOG << EOF
!date
set timing on
connect / as sysdba
startup
pfile=$KIT_DIR/init_${ORACLE_SID}.ora
!date
exit
EOF

for i in $SECONDARY_NODES
do
rsh $i -n $KIT_DIR/rstart
done

#sqlplus /NOLOG << EOF
#connect / as sysdba
#alter system set db_file_multiblock_read_count=128 sid='*';
#exit;
#EOF
```

F.15 2shut

```
#!/bin/ksh
. $FRAME_PATH/env

if [ "$1" = "abort" ]; then
for i in $SECONDARY_NODES
do
rsh $i -n $KIT_DIR/rshuta
done
sqlplus << !
connect / as sysdba
shutdown abort
exit
!
else
for i in $SECONDARY_NODES
do
rsh $i -n $KIT_DIR/rshut
done
sqlplus << !
connect / as sysdba
shutdown immediate
exit
!
fi
```

F.16 rstart

```
sqlplus /NOLOG <<EOF
!date
set timing on
connect / as sysdba
startup
pfile=$KIT_DIR/init_${ORACLE_SID}.ora
!date
exit
EOF
```

F.17 rshut

```
sqlplus /NOLOG <<!
connect / as sysdba
shutdown immediate
!
```

F.18 rc.local

```
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local

/oracle/scripts/1TB/start.hba.sh
/oracle/scripts/1TB/tuning.proc.parameters.sh
/oracle/scripts/1TB/change.mtu.sh
```

F.19 starthba.sh

```
#!/bin/sh

/sbin/modprobe qla2300

sleep 10

if [ "2.4.21-1.1931.2.399.ent" = `uname -r` ];
then

    /oracle/scripts/1TB/clear.proc.scsi.scsi.sh
    /oracle/scripts/1TB/probe.proc.scsi.scsi.sh
    /oracle/scripts/1TB/raw.bind.1TB.sh
    /oracle/scripts/1GB/raw.bind.1GB.sh

    /bin/mount -r /dev/sdah1 /oracle.data1
    /bin/mount -r /dev/sdai1 /oracle.data2
    /bin/mount -r /dev/sdaj1 /oracle.data3
    /bin/mount -r /dev/sdak1 /oracle.data4
fi
```

F.20 clear.proc.scsi.scsi.sh

```
#!/bin/sh


echo 'scsi remove-single-device 2 0 0 0' > /proc/scsi/scsi
echo 'scsi remove-single-device 2 0 1 0' > /proc/scsi/scsi
echo 'scsi remove-single-device 2 0 0 1' > /proc/scsi/scsi
echo 'scsi remove-single-device 2 0 1 1' > /proc/scsi/scsi
echo 'scsi remove-single-device 2 0 0 2' > /proc/scsi/scsi
echo 'scsi remove-single-device 2 0 1 2' > /proc/scsi/scsi
echo 'scsi remove-single-device 2 0 0 3' > /proc/scsi/scsi
echo 'scsi remove-single-device 2 0 1 3' > /proc/scsi/scsi
echo 'scsi remove-single-device 2 0 0 4' > /proc/scsi/scsi
echo 'scsi remove-single-device 2 0 1 4' > /proc/scsi/scsi
echo 'scsi remove-single-device 2 0 0 5' > /proc/scsi/scsi
echo 'scsi remove-single-device 2 0 1 5' > /proc/scsi/scsi
echo 'scsi remove-single-device 2 0 0 6' > /proc/scsi/scsi
```

```
echo 'scsi remove-single-device 2 0 1 6' > /proc/scsi/scsi
echo 'scsi remove-single-device 2 0 0 8' > /proc/scsi/scsi
echo 'scsi remove-single-device 2 0 1 8' > /proc/scsi/scsi
echo 'scsi remove-single-device 2 0 0 9' > /proc/scsi/scsi
echo 'scsi remove-single-device 2 0 1 9' > /proc/scsi/scsi

echo 'scsi remove-single-device 3 0 0 0' > /proc/scsi/scsi
echo 'scsi remove-single-device 3 0 1 0' > /proc/scsi/scsi
echo 'scsi remove-single-device 3 0 0 1' > /proc/scsi/scsi
echo 'scsi remove-single-device 3 0 1 1' > /proc/scsi/scsi
echo 'scsi remove-single-device 3 0 0 2' > /proc/scsi/scsi
echo 'scsi remove-single-device 3 0 1 2' > /proc/scsi/scsi
echo 'scsi remove-single-device 3 0 0 3' > /proc/scsi/scsi
echo 'scsi remove-single-device 3 0 1 3' > /proc/scsi/scsi
echo 'scsi remove-single-device 3 0 0 4' > /proc/scsi/scsi
echo 'scsi remove-single-device 3 0 1 4' > /proc/scsi/scsi
echo 'scsi remove-single-device 3 0 0 5' > /proc/scsi/scsi
echo 'scsi remove-single-device 3 0 1 5' > /proc/scsi/scsi
echo 'scsi remove-single-device 3 0 0 6' > /proc/scsi/scsi
echo 'scsi remove-single-device 3 0 1 6' > /proc/scsi/scsi
echo 'scsi remove-single-device 3 0 0 8' > /proc/scsi/scsi
echo 'scsi remove-single-device 3 0 1 8' > /proc/scsi/scsi
```

```
echo 'scsi remove-single-device 3 0 0 9' > /proc/scsi/scsi
echo 'scsi remove-single-device 3 0 1 9' > /proc/scsi/scsi

echo 'scsi remove-single-device 4 0 0 0' > /proc/scsi/scsi
echo 'scsi remove-single-device 4 0 1 0' > /proc/scsi/scsi
echo 'scsi remove-single-device 4 0 0 1' > /proc/scsi/scsi
echo 'scsi remove-single-device 4 0 1 1' > /proc/scsi/scsi
echo 'scsi remove-single-device 4 0 0 2' > /proc/scsi/scsi
echo 'scsi remove-single-device 4 0 1 2' > /proc/scsi/scsi
echo 'scsi remove-single-device 4 0 0 3' > /proc/scsi/scsi
echo 'scsi remove-single-device 4 0 1 3' > /proc/scsi/scsi
echo 'scsi remove-single-device 4 0 0 4' > /proc/scsi/scsi
echo 'scsi remove-single-device 4 0 1 4' > /proc/scsi/scsi
echo 'scsi remove-single-device 4 0 0 5' > /proc/scsi/scsi
echo 'scsi remove-single-device 4 0 1 5' > /proc/scsi/scsi
echo 'scsi remove-single-device 4 0 0 6' > /proc/scsi/scsi
echo 'scsi remove-single-device 4 0 1 6' > /proc/scsi/scsi
echo 'scsi remove-single-device 4 0 0 8' > /proc/scsi/scsi
echo 'scsi remove-single-device 4 0 1 8' > /proc/scsi/scsi
echo 'scsi remove-single-device 4 0 0 9' > /proc/scsi/scsi
echo 'scsi remove-single-device 4 0 1 9' > /proc/scsi/scsi
```

```
echo 'scsi remove-single-device 5 0 0 0' >
/proc/scsi/scsi
echo 'scsi remove-single-device 5 0 1 0' >
/proc/scsi/scsi
echo 'scsi remove-single-device 5 0 0 1' >
/proc/scsi/scsi
echo 'scsi remove-single-device 5 0 1 1' >
/proc/scsi/scsi
echo 'scsi remove-single-device 5 0 0 2' >
/proc/scsi/scsi
echo 'scsi remove-single-device 5 0 1 2' >
/proc/scsi/scsi
echo 'scsi remove-single-device 5 0 0 3' >
/proc/scsi/scsi
echo 'scsi remove-single-device 5 0 1 3' >
/proc/scsi/scsi
echo 'scsi remove-single-device 5 0 0 4' >
/proc/scsi/scsi
echo 'scsi remove-single-device 5 0 1 4' >
/proc/scsi/scsi
echo 'scsi remove-single-device 5 0 0 5' >
/proc/scsi/scsi
echo 'scsi remove-single-device 5 0 1 5' >
/proc/scsi/scsi
echo 'scsi remove-single-device 5 0 0 6' >
/proc/scsi/scsi
echo 'scsi remove-single-device 5 0 1 6' >
/proc/scsi/scsi
echo 'scsi remove-single-device 5 0 0 8' >
/proc/scsi/scsi
echo 'scsi remove-single-device 5 0 1 8' >
/proc/scsi/scsi
echo 'scsi remove-single-device 5 0 0 9' >
/proc/scsi/scsi
echo 'scsi remove-single-device 5 0 1 9' >
/proc/scsi/scsi
```

F.21 probe.proc.scsi.scsi.sh

```
#!/bin/sh

echo 'scsi add-single-device 2 0 0 0' >
/proc/scsi/scsi
echo 'scsi add-single-device 3 0 1 0' >
/proc/scsi/scsi
echo 'scsi add-single-device 4 0 0 0' >
/proc/scsi/scsi
echo 'scsi add-single-device 5 0 0 0' >
/proc/scsi/scsi

echo 'scsi add-single-device 2 0 1 1' >
/proc/scsi/scsi
echo 'scsi add-single-device 3 0 0 1' >
/proc/scsi/scsi
echo 'scsi add-single-device 4 0 1 1' >
/proc/scsi/scsi
echo 'scsi add-single-device 5 0 1 1' >
/proc/scsi/scsi

echo 'scsi add-single-device 2 0 0 2' >
/proc/scsi/scsi
echo 'scsi add-single-device 3 0 1 2' >
/proc/scsi/scsi
echo 'scsi add-single-device 4 0 0 2' >
/proc/scsi/scsi
echo 'scsi add-single-device 5 0 0 2' >
/proc/scsi/scsi

echo 'scsi add-single-device 2 0 1 3' >
/proc/scsi/scsi
echo 'scsi add-single-device 3 0 0 3' >
/proc/scsi/scsi
echo 'scsi add-single-device 4 0 1 3' >
/proc/scsi/scsi
echo 'scsi add-single-device 5 0 1 3' >
/proc/scsi/scsi

echo 'scsi add-single-device 2 0 0 4' >
/proc/scsi/scsi
echo 'scsi add-single-device 3 0 1 4' >
/proc/scsi/scsi
echo 'scsi add-single-device 4 0 0 4' >
/proc/scsi/scsi
echo 'scsi add-single-device 5 0 0 4' >
/proc/scsi/scsi

echo 'scsi add-single-device 2 0 1 5' >
```

```
/proc/scsi/scsi
echo 'scsi add-single-device 3 0 0 5' >
/proc/scsi/scsi
echo 'scsi add-single-device 4 0 1 5' >
/proc/scsi/scsi
echo 'scsi add-single-device 5 0 1 5' >
/proc/scsi/scsi

echo 'scsi add-single-device 2 0 0 6' >
/proc/scsi/scsi
echo 'scsi add-single-device 3 0 1 6' >
/proc/scsi/scsi
echo 'scsi add-single-device 4 0 0 6' >
/proc/scsi/scsi
echo 'scsi add-single-device 5 0 0 6' >
/proc/scsi/scsi



echo 'scsi add-single-device 2 0 1 8' >
/proc/scsi/scsi
echo 'scsi add-single-device 3 0 0 8' >
/proc/scsi/scsi
echo 'scsi add-single-device 4 0 1 8' >
/proc/scsi/scsi
echo 'scsi add-single-device 5 0 1 8' >
/proc/scsi/scsi

echo 'scsi add-single-device 2 0 0 9' >
/proc/scsi/scsi
echo 'scsi add-single-device 3 0 1 9' >
/proc/scsi/scsi
echo 'scsi add-single-device 4 0 0 9' >
/proc/scsi/scsi
echo 'scsi add-single-device 5 0 0 9' >
/proc/scsi/scsi
```

F.22 raw.bind.1TB.sh

```
#!/bin/sh



/usr/bin/raw /dev/raw/line_1    /dev/sdb1
/usr/bin/raw /dev/raw/line_2    /dev/sdc1
/usr/bin/raw /dev/raw/line_3    /dev/sdd1
/usr/bin/raw /dev/raw/line_4    /dev/sde1
/usr/bin/raw /dev/raw/line_5    /dev/sdf1
/usr/bin/raw /dev/raw/line_6    /dev/sdg1
/usr/bin/raw /dev/raw/line_7    /dev/sdh1
/usr/bin/raw /dev/raw/line_8    /dev/sdi1
/usr/bin/raw /dev/raw/line_9    /dev/sdj1
/usr/bin/raw /dev/raw/line_10   /dev/sdk1
/usr/bin/raw /dev/raw/line_11   /dev/sdl1
/usr/bin/raw /dev/raw/line_12   /dev/sdm1
/usr/bin/raw /dev/raw/line_13   /dev/sdn1
/usr/bin/raw /dev/raw/line_14   /dev/sdo1
/usr/bin/raw /dev/raw/line_15   /dev/sdp1
/usr/bin/raw /dev/raw/line_16   /dev/sdq1
/usr/bin/raw /dev/raw/line_17   /dev/sdr1
/usr/bin/raw /dev/raw/line_18   /dev/sds1
/usr/bin/raw /dev/raw/line_19   /dev/sdt1
/usr/bin/raw /dev/raw/line_20   /dev/sdu1
/usr/bin/raw /dev/raw/line_21   /dev/sdv1
/usr/bin/raw /dev/raw/line_22   /dev/sdw1
/usr/bin/raw /dev/raw/line_23   /dev/sdx1
/usr/bin/raw /dev/raw/line_24   /dev/sdy1
/usr/bin/raw /dev/raw/line_25   /dev/sdz1
/usr/bin/raw /dev/raw/line_26   /dev/sdaa1
/usr/bin/raw /dev/raw/line_27   /dev/sdab1
/usr/bin/raw /dev/raw/line_28   /dev/sdac1
/usr/bin/raw /dev/raw/line_29   /dev/sdad1
/usr/bin/raw /dev/raw/line_30   /dev/sdae1
/usr/bin/raw /dev/raw/line_31   /dev/sdaf1
/usr/bin/raw /dev/raw/line_32   /dev/sdag1

/usr/bin/raw /dev/raw/ord_1     /dev/sdb2
/usr/bin/raw /dev/raw/ord_2     /dev/sdc2
/usr/bin/raw /dev/raw/ord_3     /dev/sdd2
/usr/bin/raw /dev/raw/ord_4     /dev/sde2
/usr/bin/raw /dev/raw/ord_5     /dev/sdf2
/usr/bin/raw /dev/raw/ord_6     /dev/sdg2
/usr/bin/raw /dev/raw/ord_7     /dev/sdh2
/usr/bin/raw /dev/raw/ord_8     /dev/sdi2
/usr/bin/raw /dev/raw/ord_9     /dev/sdj2
/usr/bin/raw /dev/raw/ord_10    /dev/sdk2
/usr/bin/raw /dev/raw/ord_11    /dev/sdl2
/usr/bin/raw /dev/raw/ord_12    /dev/sdm2
/usr/bin/raw /dev/raw/ord_13    /dev/sdn2
```

/usr/bin/raw /dev/raw/ord_14 /dev/sdo2
/usr/bin/raw /dev/raw/ord_15 /dev/sdp2
/usr/bin/raw /dev/raw/ord_16 /dev/sdq2
/usr/bin/raw /dev/raw/ord_17 /dev/sdr2
/usr/bin/raw /dev/raw/ord_18 /dev/sds2
/usr/bin/raw /dev/raw/ord_19 /dev/sdt2
/usr/bin/raw /dev/raw/ord_20 /dev/sdu2
/usr/bin/raw /dev/raw/ord_21 /dev/sdv2
/usr/bin/raw /dev/raw/ord_22 /dev/sdw2
/usr/bin/raw /dev/raw/ord_23 /dev/sdx2
/usr/bin/raw /dev/raw/ord_24 /dev/sdy2
/usr/bin/raw /dev/raw/ord_25 /dev/sdz2
/usr/bin/raw /dev/raw/ord_26 /dev/sdaa2
/usr/bin/raw /dev/raw/ord_27 /dev/sdab2
/usr/bin/raw /dev/raw/ord_28 /dev/sdac2
/usr/bin/raw /dev/raw/ord_29 /dev/sdad2
/usr/bin/raw /dev/raw/ord_30 /dev/sdae2
/usr/bin/raw /dev/raw/ord_31 /dev/sdaf2
/usr/bin/raw /dev/raw/ord_32 /dev/sdag2

/usr/bin/raw /dev/raw/rest_1 /dev/sdb3
/usr/bin/raw /dev/raw/rest_2 /dev/sdc3
/usr/bin/raw /dev/raw/rest_3 /dev/sdd3
/usr/bin/raw /dev/raw/rest_4 /dev/sde3
/usr/bin/raw /dev/raw/rest_5 /dev/sdf3
/usr/bin/raw /dev/raw/rest_6 /dev/sdg3
/usr/bin/raw /dev/raw/rest_7 /dev/sdh3
/usr/bin/raw /dev/raw/rest_8 /dev/sdi3
/usr/bin/raw /dev/raw/rest_9 /dev/sdj3
/usr/bin/raw /dev/raw/rest_10 /dev/sdk3
/usr/bin/raw /dev/raw/rest_11 /dev/sdl3
/usr/bin/raw /dev/raw/rest_12 /dev/sdm3
/usr/bin/raw /dev/raw/rest_13 /dev/sdn3
/usr/bin/raw /dev/raw/rest_14 /dev/sdo3
/usr/bin/raw /dev/raw/rest_15 /dev/sdp3
/usr/bin/raw /dev/raw/rest_16 /dev/sdq3
/usr/bin/raw /dev/raw/rest_17 /dev/sdr3
/usr/bin/raw /dev/raw/rest_18 /dev/sds3
/usr/bin/raw /dev/raw/rest_19 /dev/sdt3
/usr/bin/raw /dev/raw/rest_20 /dev/sdu3
/usr/bin/raw /dev/raw/rest_21 /dev/sdv3
/usr/bin/raw /dev/raw/rest_22 /dev/sdw3
/usr/bin/raw /dev/raw/rest_23 /dev/sdx3
/usr/bin/raw /dev/raw/rest_24 /dev/sdy3
/usr/bin/raw /dev/raw/rest_25 /dev/sdz3
/usr/bin/raw /dev/raw/rest_26 /dev/sdaa3
/usr/bin/raw /dev/raw/rest_27 /dev/sdab3
/usr/bin/raw /dev/raw/rest_28 /dev/sdac3
/usr/bin/raw /dev/raw/rest_29 /dev/sdad3
/usr/bin/raw /dev/raw/rest_30 /dev/sdae3
/usr/bin/raw /dev/raw/rest_31 /dev/sdaf3
/usr/bin/raw /dev/raw/rest_32 /dev/sdag3

/usr/bin/raw /dev/raw/psupp_1 /dev/sdb5
/usr/bin/raw /dev/raw/psupp_2 /dev/sdc5
/usr/bin/raw /dev/raw/psupp_3 /dev/sdd5
/usr/bin/raw /dev/raw/psupp_4 /dev/sde5
/usr/bin/raw /dev/raw/psupp_5 /dev/sdf5
/usr/bin/raw /dev/raw/psupp_6 /dev/sdg5
/usr/bin/raw /dev/raw/psupp_7 /dev/sdh5
/usr/bin/raw /dev/raw/psupp_8 /dev/sdi5
/usr/bin/raw /dev/raw/psupp_9 /dev/sdj5
/usr/bin/raw /dev/raw/psupp_10 /dev/sdk5
/usr/bin/raw /dev/raw/psupp_11 /dev/sdl5
/usr/bin/raw /dev/raw/psupp_12 /dev/sdm5
/usr/bin/raw /dev/raw/psupp_13 /dev/sdn5
/usr/bin/raw /dev/raw/psupp_14 /dev/sdo5
/usr/bin/raw /dev/raw/psupp_15 /dev/sdp5
/usr/bin/raw /dev/raw/psupp_16 /dev/sdq5
/usr/bin/raw /dev/raw/psupp_17 /dev/sdr5
/usr/bin/raw /dev/raw/psupp_18 /dev/sds5
/usr/bin/raw /dev/raw/psupp_19 /dev/sdt5
/usr/bin/raw /dev/raw/psupp_20 /dev/sdu5
/usr/bin/raw /dev/raw/psupp_21 /dev/sdv5
/usr/bin/raw /dev/raw/psupp_22 /dev/sdw5
/usr/bin/raw /dev/raw/psupp_23 /dev/sdx5
/usr/bin/raw /dev/raw/psupp_24 /dev/sdy5
/usr/bin/raw /dev/raw/psupp_25 /dev/sdz5
/usr/bin/raw /dev/raw/psupp_26
/dev/sdaa5
/usr/bin/raw /dev/raw/psupp_27
/dev/sdab5
/usr/bin/raw /dev/raw/psupp_28
/dev/sdac5
/usr/bin/raw /dev/raw/psupp_29
/dev/sdad5
/usr/bin/raw /dev/raw/psupp_30
/dev/sdae5

/usr/bin/raw /dev/raw/psupp_31 /dev/sdaf5
/usr/bin/raw /dev/raw/psupp_32 /dev/sdag5

/usr/bin/raw /dev/raw/index_1 /dev/sdb6
/usr/bin/raw /dev/raw/index_2 /dev/sdc6
/usr/bin/raw /dev/raw/index_3 /dev/sdd6
/usr/bin/raw /dev/raw/index_4 /dev/sde6
/usr/bin/raw /dev/raw/index_5 /dev/sdf6
/usr/bin/raw /dev/raw/index_6 /dev/sdg6
/usr/bin/raw /dev/raw/index_7 /dev/sdh6
/usr/bin/raw /dev/raw/index_8 /dev/sdi6
/usr/bin/raw /dev/raw/index_9 /dev/sdj6
/usr/bin/raw /dev/raw/index_10 /dev/sdk6
/usr/bin/raw /dev/raw/index_11 /dev/sdl6
/usr/bin/raw /dev/raw/index_12 /dev/sdm6
/usr/bin/raw /dev/raw/index_13 /dev/sdn6
/usr/bin/raw /dev/raw/index_14 /dev/sdo6
/usr/bin/raw /dev/raw/index_15 /dev/sdp6
/usr/bin/raw /dev/raw/index_16 /dev/sdq6
/usr/bin/raw /dev/raw/index_17 /dev/sdr6
/usr/bin/raw /dev/raw/index_18 /dev/sds6
/usr/bin/raw /dev/raw/index_19 /dev/sdt6
/usr/bin/raw /dev/raw/index_20 /dev/sdu6
/usr/bin/raw /dev/raw/index_21 /dev/sdv6
/usr/bin/raw /dev/raw/index_22 /dev/sdw6
/usr/bin/raw /dev/raw/index_23 /dev/sdx6
/usr/bin/raw /dev/raw/index_24 /dev/sdy6
/usr/bin/raw /dev/raw/index_25 /dev/sdz6
/usr/bin/raw /dev/raw/index_26 /dev/sdaa6
/usr/bin/raw /dev/raw/index_27 /dev/sdab6
/usr/bin/raw /dev/raw/index_28 /dev/sdac6
/usr/bin/raw /dev/raw/index_29 /dev/sdad6
/usr/bin/raw /dev/raw/index_30 /dev/sdae6
/usr/bin/raw /dev/raw/index_31 /dev/sdaf6
/usr/bin/raw /dev/raw/index_32 /dev/sdag6

/usr/bin/raw /dev/raw/temp_1 /dev/sdb7
/usr/bin/raw /dev/raw/temp_2 /dev/sdc7
/usr/bin/raw /dev/raw/temp_3 /dev/sdd7
/usr/bin/raw /dev/raw/temp_4 /dev/sde7
/usr/bin/raw /dev/raw/temp_5 /dev/sdf7
/usr/bin/raw /dev/raw/temp_6 /dev/sdg7
/usr/bin/raw /dev/raw/temp_7 /dev/sdh7
/usr/bin/raw /dev/raw/temp_8 /dev/sdi7
/usr/bin/raw /dev/raw/temp_9 /dev/sdj7
/usr/bin/raw /dev/raw/temp_10 /dev/sdk7
/usr/bin/raw /dev/raw/temp_11 /dev/sdl7
/usr/bin/raw /dev/raw/temp_12 /dev/sdm7
/usr/bin/raw /dev/raw/temp_13 /dev/sdn7
/usr/bin/raw /dev/raw/temp_14 /dev/sdo7
/usr/bin/raw /dev/raw/temp_15 /dev/sdp7
/usr/bin/raw /dev/raw/temp_16 /dev/sdq7
/usr/bin/raw /dev/raw/temp_17 /dev/sdr7
/usr/bin/raw /dev/raw/temp_18 /dev/sds7
/usr/bin/raw /dev/raw/temp_19 /dev/sdt7
/usr/bin/raw /dev/raw/temp_20 /dev/sdu7
/usr/bin/raw /dev/raw/temp_21 /dev/sdv7

```
/usr/bin/raw /dev/raw/temp_22 /dev/sdw7
/usr/bin/raw /dev/raw/temp_23 /dev/sdx7
/usr/bin/raw /dev/raw/temp_24 /dev/sdy7
/usr/bin/raw /dev/raw/temp_25 /dev/sdz7
/usr/bin/raw /dev/raw/temp_26 /dev/sdaa7
/usr/bin/raw /dev/raw/temp_27 /dev/sdab7
/usr/bin/raw /dev/raw/temp_28 /dev/sdac7
/usr/bin/raw /dev/raw/temp_29 /dev/sdad7
/usr/bin/raw /dev/raw/temp_30 /dev/sdae7
/usr/bin/raw /dev/raw/temp_31 /dev/sdaf7
/usr/bin/raw /dev/raw/temp_32 /dev/sdag7

/usr/bin/raw /dev/raw/undo_1 /dev/sdb8
/usr/bin/raw /dev/raw/undo_2 /dev/sdc8
/usr/bin/raw /dev/raw/undo_3 /dev/sdd8
/usr/bin/raw /dev/raw/undo_4 /dev/sde8

/usr/bin/raw /dev/raw/log_1 /dev/sdf8
/usr/bin/raw /dev/raw/log_2 /dev/sdg8
/usr/bin/raw /dev/raw/log_3 /dev/sdh8
/usr/bin/raw /dev/raw/log_4 /dev/sdi8
/usr/bin/raw /dev/raw/log_5 /dev/sdj8
/usr/bin/raw /dev/raw/log_6 /dev/sdk8
/usr/bin/raw /dev/raw/log_7 /dev/sdl8
/usr/bin/raw /dev/raw/log_8 /dev/sdm8

/usr/bin/raw /dev/raw/sys_1 /dev/sdn8

/usr/bin/raw /dev/raw/sysaux_1 /dev/sdo8

/usr/bin/raw /dev/raw/cntrl_1 /dev/sdp8
/usr/bin/raw /dev/raw/cntrl_2 /dev/sdq8

/usr/bin/raw /dev/raw/default_1 /dev/sdr8

/usr/bin/raw /dev/raw/ocr_1 /dev/sds8

/usr/bin/raw /dev/raw/quorum_1 /dev/sdt8

/usr/bin/raw -qa
```

F.23 tuning.proc.parameters.sh

```
#/bin/ksh

/sbin/sysctl -w fs.aio-max-size=8388608

/sbin/sysctl -w net.core.rmem_max=1048576
/sbin/sysctl -w net.core.wmem_max=1048576
/sbin/sysctl -w
net.core.rmem_default=1048576
/sbin/sysctl -w
net.core.wmem_default=1048576
```

F.24 change.mtu.sh

```
#/bin/ksh

/sbin/ifconfig eth1 mtu 9000
/sbin/ifconfig eth2 mtu 9000
```

Appendix G Pricing Information

-----Original Message-----
From: MaryBeth Pierantoni [mailto:mary.beth.pierantoni@oracle.com]
Sent: Monday, October 18, 2003 6:44 PM
To: FengRui@lenovo.com
Cc: mary.beth.pierantoni@oracle.com, Ray.Glasstone@oracle.com
Subject: Oracle Pricing

Product	**Price (RMB)**	**Quantity**	**Extended Price (RMB)**
Oracle Database 10g Enterprise Edition for 3 years, Named User Plus		16	1,324,200
Real Application Clusters for 3 years, Named User Plus		16	662,200
Partitioning for 3 years, Named User Plus		16	331,000
Oracle Database Server Support Package for 3 years		3	198,648
Oracle Mandatory E-Business Discount			(503,210)
TOTAL (RMB)			**2012838**

For Oracle pricing contact:
MaryBeth Pierantoni
+1- (1)650-506-2118
mary.beth.pierantoni@oracle.com

For Legend pricing comtact:
Yajie Li
+86-010-62988888-1342
liyjg@lenovo.com