Compaq Computer Corporation

ProLiant 6000-6/200 Model -1X
Oracle V7.3.3

TPC Benchmark™D
Full Disclosure Report

First Edition
May 1997

First Edition – May 2, 1997

Compaq Computer Corporation, as the Sponsor of this benchmark test, believes that the information in this document is accurate as of the publication date. The information in this document is subject to change without notice. The Sponsor assumes no responsibility for any errors that may appear in this document.

The pricing information in this document is believed to accurately reflect the current prices as of the publication date. However, the Sponsor provides no warranty of the pricing information in this document.

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

All performance data contained in this report was obtained in a rigorously controlled environment. Results obtained in other operating environments may vary significantly. No warranty of system performance or price/performance is expressed or implied in this report.

Printed in the United States May 2, 1997

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark™ D test conducted on the ProLiant 6000-6/200 using Oracle7 Server™, in conformance with the requirements of the TPC Benchmark™ D Standard Specification, Revision 1.2. The operating system used for the benchmark was Microsoft Windows NT 4.0 (build 1381 – Service Pack 1). The application was written in C and compiled using Microsoft Visual C++ for Windows NT.

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

Standard and Executive Summary Statements

Pages ii-iv contain the Executive Summary and Numerical Quantities Summary of the benchmark results for the Compaq ProLiant 6000-6/200 Model-1X.

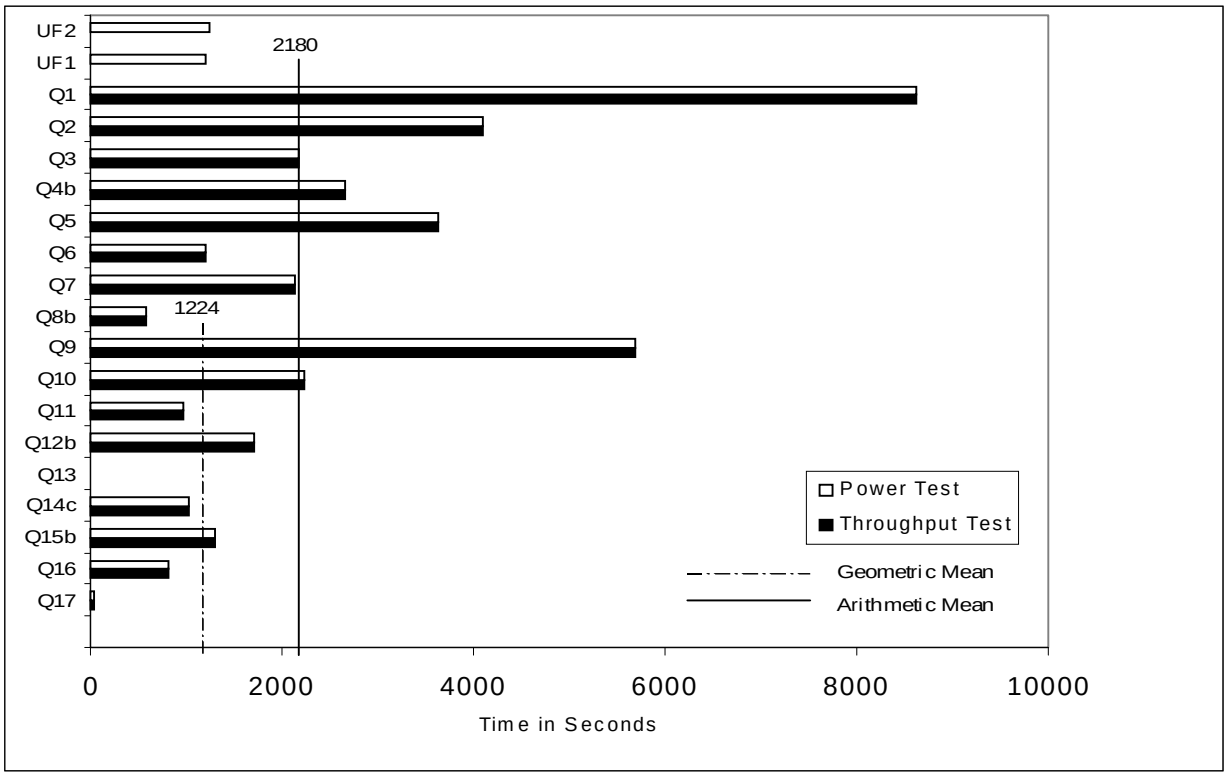
Auditor

The benchmark configuration, environment and methodology used to produce and validate the test results, and the pricing model used to calculate the cost per QppD and QthD were audited by Lorna Livingtree of Performance Metrics, Inc. to verify compliance with the relevant TPC specifications. The auditor's letter of attestation is attached as Appendix F.

COMPAQ COMPUTER CORPORATION	ProLiant 6000 MODEL-1X	TPC-D 1.2.1
		REPORT DATE: 2-May-97

TOTAL SYSTEM COST	TPC-D POWER	TPC-D THROUGHPUT	PRICE/PERFORMANCE
$ 427,580	**294.2** QppD @ 100GB	**147.8** QthD @ 100GB	**2,051** $/per QphD @ 100GB

DATABASE SIZE	DATABASE MANAGER	OPERATING SYSTEM	OTHER SOFTWARE	AVAILABILITY DATE
100GB	Oracle Version 7.3.3	Microsoft Windows NT 4.0	MKS Toolkit Resource Kit Visual C++	June 30, 1997



DATABASE LOAD TIME = 24 HOURS 7 MINUTES 48 SECONDS	DISK SIZE/DATABASE SIZE = 4.95	RAID: N

SYSTEM COMPONENTS	DESCRIPTION
NUMBER OF NODES	COMPAQ PROLIANT 6000-6/200 – MODEL-1X (1 NODE)
PROCESSORS	4 X 200MHZ PENTIUMPRO®, 512K CACHE
MEMORY	2 GB
DISK DRIVES	58 X 9.1 GB AND 1 X 4.3 GB
TOTAL GB OF STORAGE	495 GB

Compaq Computer Corporation	Proliant 6000 - 6/200 Model - 1X			Report Date:	2-May-97
Description	**Part Number**	**Unit Price**	**Qty**	**Extended Price**	**5 yr. Maint. Price**
Server Hardware					
Proliant 6000 6/200 512K Model 1X	273350-001	14,337	1	14,337	5,018
PentiumPro/200MHz Processor					
Netelligent 10/100 UTP PCI NIC					
2 x Integrated Fast-SCSI-2 Controller					
CD-ROM.SmartStart,Configuration Utility					
Proliant 6000 Internal Drive Cage	273314-B21	432	1	432	151
Rack External Keyboard	187344-001	32	1	32	11
Keyboard/Mouse/Monitor Extension Cables	169989-001	76	1	76	27
42U 19" Rack	165753-001	1,620	1	1,620	567
22U 19" Rack	163747-001	1,458	1	1,458	510
Rack Sidewall Kit (For 42U Rack)	165652-001	201	1	201	70
Proliant 6000 Rack Mount Kit	273316-B21	492	1	492	172
6/200 - 512K Processor Expansion Kit	273316-B21	2,905	3	8,715	3,050
1 GB DIMM Kit	241774-B21	26,460	2	52,920	18,522
SMART-2/P SCSI Array Controller	194753-001	2,138	5	10,692	3,742
Compaq VGA V50 - 15" Color Monitor	264150-001	383	1	383	134
Compaq UPS M2500	163760-001	1,446	2	2,892	1,012
			Subtotal	**94,250**	**32,987**
Server Software					
Oracle 7.3.3 Enterprise Database Server for Windows NT		35,880	1	35,880	48,960
MKS Toolkit for NT		499	1	499	5 Yr WTY
Microsoft Visual C++ and Subscription Service		499	1	499	2,495
Microsoft Windows NT Server v. 4.0		809	1	809	4,045
Microsoft Resource Kit for NT 4.0		499	1	499	Inc Above
			Subtotal	**38,186**	**55,500**
Storage Devices					
Rack-Mountable Proliant Storage System -F1	272800-001	1,620	8	12,960	4,536
- Includes all cables and connectors					
4.3 GB Pluggable SCSI-2 Drive	146742-006	1,227	1	1,227	429
9.1 GB Pluggable SCSI-2 Drive	199882-001	2,376	58	137,808	48,233
4\16GB TurboDAT Drive	142181-001	1,084	1	1,084	379
			Subtotal	**153,079**	**53,578**
			Total	**$285,515**	**$142,065**
			Five-Year Cost of Ownership:		**$427,580**
Audited By: Lorna Livingtree of Performance Metrics, Inc.					

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

Compaq Computer Corporation	Numerical Quantities	TPC-D Rev. 1.2.1 Report Date: 2-May-97

Measurement Results:

Database Scaling (SF/Size)	=	100
Total Data Storage/Database Size	=	4.95
Database Load Time	=	24 hr. 7 min 48 sec
Query Streams for Throughput Test (S)	=	0
TPC-D Power Metric (QppD@100GB)	=	294.2
TPC-D Throughput Metric (QthD@100GB)	=	147.8
Total System Price Over five years	=	$ 427,580
TPC-D Power Throughput Metric ($/QphD@100GB)	=	2,051

Measurement Intervals:

Measurement interval in Performance Test (TS) = 41,403 s

Duration of Stream Execution:

Stream ID	Seed	Start Date	Start Time	End Date	End Time	Total Time
Stream00	697726	4/23/97	00:11:33	4/23/97	11:41:36	11:30:03
UF1		4/23/97	00:11:33	4/23/97	00:31:33	
UF2		4/23/97	11:20:56	4/23/97	11:41:36	

TPC-D Timing Intervals (in seconds):

Query	1	2	3	4	5	6	7	8	9
Stream 00	8624.0	4087.5	2183.3	2656.8	3632.4	1211.9	2145.5	585.7	5687.7

Query	10	11	12	13	14	15	16	17	UF1	UF2
Stream 00	2232.8	972.9	1713.4	24.4	1030.8	1300.9	823.3	48.4	1199.4	1240.2

Table Of Contents

General Items

1.1 Test Sponsor

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

Compaq Computer Corporation is the sponsor of this TPC Benchmark™ D.

1.2 Parameter Settings

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

- *Database Tuning Options*
- *Optimizer/Query execution options*
- *Query processing tool/language configuration parameters*
- *Recovery/commit options*
- *Consistency/locking options*
- *Operating system and configuration parameters*

- *Configuration parameters and options for any other software component incorporated into the pricing structure*
- *Compiler optimization options*

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

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

1.3 Configuration Items

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

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

The server System Under Test (SUT), a Compaq ProLiant 6000-6/200, depicted in Figure 1.1, consisted of:

Four 200 MHz PentiumPro® Processors.
2048 MB of Memory.
Four SMART-2 SCSI Controllers with 14x9.1GB Drives.
One SMART-2 SCSI Controller with 2x9.1GB Drives.
One 4.3GB Fast/Wide SCSI-2 Drive.
Netelligent 10/100 UPT PCI Network Card
One 4/16 Turbo-DAT drive.

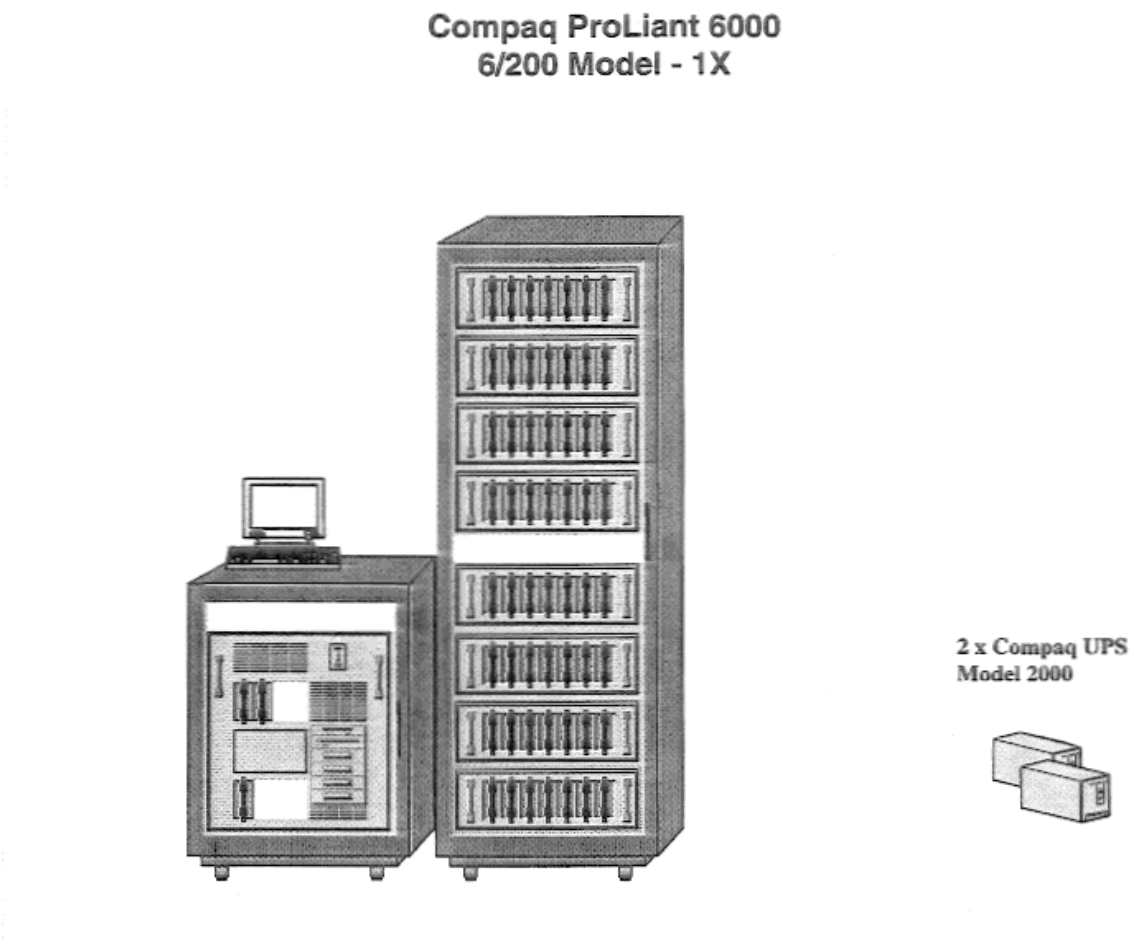


Figure 1. 1 Benchmarked and Priced Configuration

Clause 1: Logical Database Design

2.1 Table Definitions

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

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

2.2 Physical Organization of Database

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

No record clustering or index clustering was used.

2.3 Horizontal Partitioning

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

The database layout is described in the table in sections 5.2 and 5.3.

2.4 Replication

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

Replication was not used for this benchmark.

Clause 2: Queries and Update Functions related items

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 Random Number Generation

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

The 1.1.0A version of DBGEN and version 1.20 of QGEN were used to generate the random numbers for this TPC-D benchmark.

3.3 Substitution Parameters Generation

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

The 1.10B version of QGEN was used to generate the substitution parameters.

3.4 Query Text and Output Data from Database

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

Appendix C contains the query text and query output

The minor query modifications used in this implementation include the following.
Wherever there is a date given in the WHERE clause of a select the TO_DATE function is used to convert the character input to the proper date format.
Wherever there is a date returned by the query the TO_CHAR function is used to convert the date format to the proper output format.
In some queries aliases have been assigned to the table names in the subqueries so that the columns are no longer ambiguous.
The Order table defined in the specification has been named the orders table due to conflict with a reserved word.
Whenever there is a time modification (i.e. add 3 months to a date) the ADD_MONTHS function is used to calculate and adjust the time delta and keep it in the date/time format.
CREATE TABLE statements have specific storage parameters in them. Namely, what tablespace to create them on, the sizes of the extents, how full the extents should be before allocating a new one, and the degree of parallelism that the table will be accessed with.

3.5 Query Substitution Parameters and Seeds Used

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

Appendix D contains the seed and query substitution parameters.

3.6 Update Function Source Code

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

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

3.7 Database Maintenance Option

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

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

Clause 3: Database System Properties

4.1 Atomicity

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

4.1.1 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 ORDER, LINEITEM, and HISTORY tables.

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

4.1.2 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 ORDER, LINEITEM, and HISTORY tables.

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

4.2 Consistency

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

4.2.1 Consistency Test

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

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

4.3 Isolation

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

4.3.1 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 completed and did not see the uncommitted changes made by the ACID transaction.
3. The ACID transaction was committed.

4.3.2 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 completed and did not see the uncommitted changes made by the ACID transaction.
3. The ACID transaction was rolled back.

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

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

4.4 Durability

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

4.4.1 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-D database tables or recovery log tables.

The disks containing the database logs were RAID-1. The tables for the database were on RAID-0 stripes.

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

4.4.2 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 test and the memory failure test were combined.

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

4.4.3 Memory Failure

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

See the previous section (section 4.4.2).

Clause 4: Scaling and Database Population

5.1 Initial Cardinality of Tables

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

Table 5. 1: Initial number of Rows

Table	Occurrences
Orders	150,000,000
Lineitem	600,037,902
Customer	15,000,000
Part	20000000
Supplier	1000000
Partsupp	80000000
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….

The database tables are distributed across 4 SMART-2 SCSI Array Controllers. All four controllers have 14 drives with a single RAID-0 stripe across it. This stripe is where the database datafiles are stored. The fifth controller has 2 drives with RAID-1. It has two logical drives for the log drives. The Flat files, for loading the database, were on the sixth controller. The operating system and the Oracle binaries were stored on a single 4.3GB disk drive on the embedded Fast/Wide SCSI-2 controller.

Table 5.2: Distribution of Tables Across Media

Controller	Disk Drive	Description of Content
1	1,2	Mirrored pair of disk drives…containing 2 log drives
2	1-14	1 RAID-0 Stripe for the database datafiles.
3	1-14	1 RAID-0 Stripe for the database datafiles.
4	1-14	1 RAID-0 Stripe for the database datafiles.
5	1-14	1 RAID-0 Stripe for the database datafiles.

5.3 Partitioning and Replication

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

No replication was used for this implementation.

Table 5.3: Distribution of Partitioned Tables

Controller	Configuration	Usage	Fraction
1	Mirrored	Log1	100%
	Mirrored	Log2	100%
2	RAID-0	lineitem, orders, parts, partsupp, supplier, customer and region	1/8th of each of these tables except region 100%
	RAID-0	lineitem, orders, parts, partsupp, supplier, customer and nation	1/8th of each of these tables except nation 100%
3	RAID-0	lineitem, orders, parts, partsupp, supplier and customer	1/8th of each of these tables
	RAID-0	lineitem, orders, parts, partsupp, supplier and customer	1/8th of each of these tables
4	RAID-0	lineitem, orders, parts, partsupp, supplier and customer	1/8th of each of these tables
	RAID-0	lineitem, orders, parts, partsupp, supplier and customer	1/8th of each of these tables
5	RAID-0	lineitem, orders, parts, partsupp, supplier and customer	1/8th of each of these tables
	RAID-0	lineitem, orders, parts, partsupp, supplier and customer	1/8th of each of these tables

5.4 DBGEN Version and Modifications

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

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

5.5 Database Content of the First Ten Rows

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

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

5.6 Database Load time

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

Database load time was 24 hours, 7 minutes and 48 seconds.

5.7 Data Storage Ratio

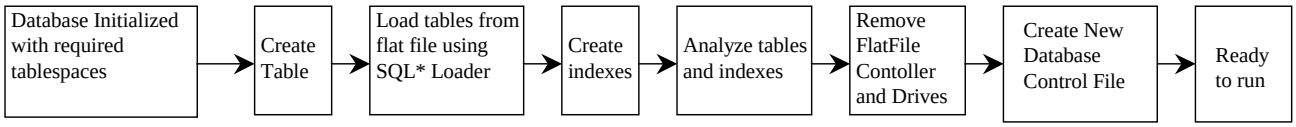
The data storage ratio must be disclosed. It is computed by dividing the total data storage of the priced configuration (expressed in GB) by the size chosen for the test database as defined in 4.1.3.1. The ratio must be reported to the nearest 1/100th, rounded up.

Disk Type	# of Disks	Space per Disk	Sub-total Disk Space	Data Storage Ratio
4.3GB	1	3.99	3.99	
9.1GB	58	8.469	491.202	
		TOTAL	**495.192**	**4.95**

5.8 Database Load Mechanism Details and Illustration

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

DBGEN was used to create flat files which were then loaded into the table spaces using SQL*Loader. The flat files were split into 40 files for the LINEITEM table, 16 files for ORDER table, 8 files for the rest except the REGION and NATION tables which each had only one file. PERL was used to perform the creation of the tablespaces, tables, and indexes of the database. It was also used to run the analyze scripts.
The controller cache was set to 75/25 read/write during the database load and during the power runs. This was done manually, through running of the Compaq Array Configuration Utility before the start of the load. After the load phase was completed, the controller and the 14 drives storing the flatfiles were physically removed from the database. On removing the controller and drives, the PHYSICALDRIVE numbers for the log files changed. So, the database control file was recreated to reflect the new PHYSICALDRIVE number associated with each log file. A time stamp was taken before starting this process which involved creation of new control file, shutting down the database and finally restarting the database. After the database was up and running, another time stamp was taken to show the amount of time that this process took. This time was counted in the load time number shown in the Numerical Quantities section of this document.


Database Initialized with required tablespaces
Create Table
Load tables from flat file using SQL* Loader
Create indexes
Analyze tables and indexes
Remove FlatFile Contoller and Drives
Create New Database Control File
Ready to run

Clause 5: Performance Metrics and Execution rules Related Items

6.1 Steps in the Power Test

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

The following steps were used to implement the power test:

1. Database Restart
2. UF1 Update Transaction
3. Stream 00 Execution
4. UF2 Update Transaction

6.2 Timing Intervals

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

The power test timing intervals are presented in the Numerical Quantities Summary in the Preface of this document.

6.3 Number of Streams for The Throughput Test

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

The throughput test used only one execution stream. Since only one stream was used, the values for the power test are used for the throughput test.

6.4 Start/Finish Time of Each Query Stream

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

The throughput test start time and finish time for each stream are reflected in the Numerical Quantities Summary in the Preface of this document.

6.5 Total Elapsed Time

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

The total elapsed time of the throughput test was 41,403 seconds.

6.6 Start/Finish Time for Update Function

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

The start and finish time for each update function in the update stream are contained in the Numerical Quantities Summary in the Preface of this document.

6.7 Timing Intervals for Each Query and Each Update

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

The timing intervals for each query and each update function, for the throughput test, are contained in the Numerical Quantities Summary in the Preface of this document. The values for the power test are used for the throughput test.

6.8 Performance Metrics

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

The performance metrics, and the numbers on which they are based, are contained in the Numerical Quantities Summary in the Preface of this document.

6.9 Reproducibility Method

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

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

Run	**QppD@100GB**	QthD@100GB	**QphD@100GB**
Run 1	296.1	149.2	210.2
Run 2	294.2	147.8	208.5
Difference	1.98	1.4	1.7

Clause 6: SUT and Driver Implementation Related Items

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.

QGEN is first called with a stream ID of 0 to generate the QET for the power test. The power test is performed by a shell script called **runpower1.sh.** This shell script calls **runuf1.sh** which performs the update function. Queries are then executed with the **qexec.exe** ISL program. The last part of the power test is run through another shell script called **runuf2.sh.** Both wall-clock and high-resolution times are collected for the measurement intervals.
Update functions are parallelized by partitioning the data set and utilizing multiple processes to perform the updates. Two Pro*C programs are written to perform the inserts and deletes required by UF1 and UF2. Batch array inserts are used for UF1 and logical consistency is ensured by limiting the batching of ORDER and LINEITEM rows to ones that have the same order key ranges.

7.2 Implementation Specific Layer (ISL)

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

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

Clause 7: Pricing Related Items

8.1 Hardware and Software Used in the Priced System

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

A detailed list of the hardware and software used in the priced system is included in the pricing sheet in the executive summary.

8.2 Total Five year Price

The total 5-year price of the entire configuration must be reported including: hardware, software, and maintenance charges. Separate component pricing is recommended. The basis of all discounts used must be disclosed.

The total 5-year price of the configuration is $427,580.

A detailed price sheet of all the hardware and software used in this configuration and the 5-year maintenance cost is included in the executive summary at the beginning of this document.

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 availability date reported on the executive summary must be the date by which all components are committed to being available. The full disclosure report must report availability dates individually for at least each of the categories for which a pricing subtotal must be provided..

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

Clause 8: Related Items

9.1 Auditor's Report & 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 D was audited by Lorna Livingtree of Performance Metrics, Inc. Further information regarding the audit process may be obtained from:

Performance Metrics, Inc.
2229 Benita Dr. Suite 101
Rancho Cordova, CA
(phone) 916/635-2822
(fax) 916/858-0109

Requests for this TPC Benchmark D Full Disclosure Report should be sent to:

Transaction Processing Performance Council
c/o Shanley Public Relations
777 North First Street, Suite 6000
San Jose, CA 95112-6311
408/295-8894

PERFORMANCE METRICS INC.
TPC Certified Auditors

May 2, 1997

Mr. Ajay Manchepalli
Database Engineer, Database Performance Engineering
Compaq Computer Corporation
20555 SH249
Houston, TX 77070

I have verified the TPC Benchmark™ D for the following configuration:

Platform: Compaq ProLiant 6000 6/200 Model-1X
Database Manager: Oracle7 Server™ version 7.3.3
Operating System: Microsoft Windows NT Server version 4.0 Service Pack 1

CPU's	Memory	Disks
4 PentiumPro @ 200 MHz	Main: 2048 MB Cache: 512 KB	1 @ 4.3GB 58 @ 9.1GB
QppD@100GB	**QthD@100GB**	**QphD@100GB**
294.2	147.8	208.5

In my opinion, these performance results were produced in compliance with the TPC requirements for the benchmark. The following attributes of the benchmark were given special attention:

- The input variables were generated by QGEN.
- The database was populated using DBGEN.
- The database was maintained by the reset method.
- The database tables were properly sized and populated.
- The database was properly scaled for scale factor 100.
- The TIME table was not used.
- The database load time was verified to be measured correctly.
- The executable query text was verified to be compliant.

2229 Benita Dr. Suite 101, Rancho Cordova, CA 95670
(916) 635-2822 fax: (916) 858-0109 email: Lorna@PerfMetrics.com

PERFORMANCE METRICS INC.
TPC Certified Auditors

- The ACID properties were verified.
- The power test was correctly measured.
- The power metric was correctly calculated.
- The ratio between the longest and the shortest query required no adjustment.
- The throughput metric was computed using the results from the power test.
- The power test was repeated and results were within 5%.
- The implementation specific layer was verified to be compliant.
- The database logs were verified to have at least 8 hours of space configured.
- Third party quotes were verified for compliance.
- The Executive Summary configuration and load diagram pages of the FDR were verified for accuracy.

Sincerely,

Lorna Livingtree

Lorna Livingtree
Auditor

Appendix A: Parameter Settings

P_BUILD.ORA

```
# Init.ora include file created by bumpx.pl
always_anti_join                = hash
db_name                 = tpcd
#control_files          = /oracle/v7/dbs/cntrltpcd.dbf
db_block_buffers        = 2000
shared_pool_size        = 3500000
parallel_max_servers    = 192
#parallel_min_servers    = 80
parallel_min_servers    = 0
parallel_server_idle_time = 2
max_dump_file_size      = 5000
audit_trail             = FALSE
global_names            = FALSE
commit_point_strength   = 1
dblink_encrypt_login    = true
db_block_size           = 8192
db_file_multiblock_read_count = 16
db_block_lru_latches    = 8
dml_locks               = 500
processes               = 1024
sessions                = 1024
transactions            = 1024
enqueue_resources       = 10240
transactions_per_rollback_segment = 10
distributed_transactions  = 20
log_checkpoint_interval = 9999999999
db_files                = 1000
open_cursors            = 2000
max_rollback_segments   = 128
#compatible                     = 7.3.2
#sort_area_size         = 10240000
#sort_area_size         =  5242880
sort_area_size          =  1048576
sort_read_fac                   = 16
sort_direct_writes      = AUTO
sort_write_buffer_size  = 65536
sort_write_buffers      = 8
#optimizer_percent_parallel = 50
#optimizer_percent_parallel = 0
hash_area_size          = 20971520
hash_multiblock_io_count = 1
################################################
compatible              = 7.3.2
control_files           = c:\tpcd\dbs\hundgig\cntrl.dbf
background_dump_dest            = c:\tpcd\log\hundgig
#optimizer_search_limit         = 7
user_dump_dest                                  = c:\tpcd\log\hundgig
```

P_LOAD.ORA

```
# Init.ora include file created by bumpx.pl
always_anti_join                = hash
db_name                 = tpcd
db_block_buffers        = 20000
#db_block_buffers        = 25000
shared_pool_size        = 250000000
parallel_server_idle_time = 1
max_dump_file_size      = 5000
audit_trail             = FALSE
global_names            = FALSE
db_block_size           = 8192
db_file_multiblock_read_count = 32
db_block_lru_latches    = 12
dml_locks               = 200
processes               = 512
sessions                = 512
transactions            = 1024
enqueue_resources       = 20480
transactions_per_rollback_segment = 1
distributed_transactions  = 10
log_checkpoint_interval = 9999999999
log_buffer              = 2621440
db_files                = 1000
open_cursors            = 512
cursor_space_for_time   = FALSE  # used to be TRUE
max_rollback_segments   = 128
sort_area_size          = 1048576
sort_read_fac                   = 15
sort_direct_writes      = AUTO
sort_write_buffer_size  = 65536
sort_write_buffers      = 8
optimizer_percent_parallel = 90
hash_area_size          = 20971520
hash_multiblock_io_count = 2
################################################
compatible                      = 7.3.3
control_files           = c:\tpcd\dbs\hundgig\cntrl.dbf
background_dump_dest            = c:\tpcd\log\hundgig
optimizer_search_limit          = 7
parallel_max_servers    = 40
parallel_min_servers    = 40
user_dump_dest                                  = c:\tpcd\log\hundgig
v733
```

P_ANALYZE.ORA

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

P_RUNF.ORA

```
always_anti_join        = hash
db_name                 = tpcd
parallel_server_idle_time =  1
max_dump_file_size      = 5000
audit_trail             = FALSE
global_names            = FALSE
db_block_size           = 8192
db_file_multiblock_read_count = 7
dml_locks               = 200
enqueue_resources       = 20480
transactions_per_rollback_segment = 1
log_checkpoint_interval = 9999999999
log_buffer              = 2621440
db_files                = 1000
open_cursors            = 256
cursor_space_for_time   = FALSE  # used to be TRUE
max_rollback_segments   = 128
sort_area_size          = 1048576
sort_read_fac           = 15
sort_direct_writes      = TRUE
sort_write_buffer_size  = 65536
sort_write_buffers      = 8
hash_multiblock_io_count = 7
compatible              = 7.3.2
control_files           = c:\tpcd\dbs\hundgig\cntrl.dbf
background_dump_dest    = c:\tpcd\log\hundgig
db_block_buffers        = 2000
db_block_lru_latches    = 4
hash_area_size          =  20971520
log_simultaneous_copies  = 8
optimizer_percent_parallel = 100
optimizer_search_limit   = 7
```

```
parallel_max_servers      = 32
parallel_min_servers      = 32
processes                 = 64
sessions                  = 96
transactions              = 128
shared_pool_size          = 88000000
user_dump_dest            = c:\tpcd\log\hundgig
v733_plans_enabled        = TRUE
```

ORACLE REGISTRY

The batch file CREATE_INSTANCE.BAT made changes to the NT Registry.

ORACLE_SID:REG_SZ:TPCD

Below is the batch file.

CREATE_INSTANCE.BAT

```
oradim73 -new -sid tpcd -intpwd internal
```

COMPAQ DEVICE DRIVERS FOR WINDOWS NT

The following Microsoft Windows NT device drivers were replaced with Compaq-specific device drivers:

The Microsoft SMART-2 Array Controller default device driver (CPQARRAY.SYS) was replaced with the Compaq SMART-2 Array Controller ECI Driver for Microsoft Windows NT 4.0 (CPQSMRT2.SYS). This driver is available on the Compaq Computer Corporation World Wide Web site (www.compaq.com). It is distributed via Softpaq SP1529.

Appendix B: Table Definitions

TPCD100_8.CONF

```
# Configuration file for bumpx.pl
#    TPCD100_40.CONF
# Digital Prioris Server configuration:
#    4CPUs, 2GB Memory, 146*4GB disks, 7 controllers, Windows NT 4.0, Oracle 7.3.3
# TPCD SF=100.
# Oracle striping of tables, indexes, temporary with 40 files.
#
*matchon
##############################################################################
# preprocessing-like directives

%b-preproc

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

*load
\sqlldr73 {}

%e-preproc
*matchoff

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

### general configuration parameters

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

### Enable one of the following for creating table files or temp/index files
### in two steps if there is not enough disk space to keep the flat files.
###
### Use the following configuration to create data tablesapces
#skip_ts=index,temp
### Use the following configuration to create index and temp tablesapces
#skip_ts=data
#skip_create_tables=true
#skip_create_user=true
###

compatible=7.3

### init.ora parameters

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

### database creation parameters

db_maxdatafiles=1020

### system/tablespace parameters

ts_def_area = c:\tpcd\dbs\hundgig\,d:\tpcd\dbs\hundgig\,e:\tpcd\dbs\hundgig\,&
f:\tpcd\dbs\hundgig\,g:\tpcd\dbs\hundgig\,h:\tpcd\dbs\hundgig\,&
i:\tpcd\dbs\hundgig\,j:\tpcd\dbs\hundgig\

ts_def_#files = 8

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

ts_log_first_size=1900m
ts_log_size=1900m
#ts_log_area=e:\tpcd\dbs\hundgig\
#ts_log_datafiles=log1.dbf,log2.dbf
ts_log_area=none
ts_log_datafiles=\\.\PHYSICALDRIVE9,\\.\PHYSICALDRIVE10
ts_log_#files_pt=2

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

# one tablespace per TPCD large object

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

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

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

ts_orders_size=2350m
ts_orders_storage=(initial 8k next 1100m pctincrease 0)

ts_parts_size=455m
ts_parts_storage=(initial 8k next 453m pctincrease 0)

ts_partsupp_size=1805m
ts_partsupp_storage=(initial 8k next 1803m pctincrease 0)

ts_customer_size=455m
ts_customer_storage=(initial 8k next 453m pctincrease 0)

ts_supplier_size=50m
ts_supplier_storage=(initial 8k next 48m pctincrease 0)
ts_supplier_coalesce_alter=1

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

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

ts_l_ored_#files=8
ts_l_ored_size=3171m
ts_l_ored_storage=(initial 8k next 100m maxextents unlimited pctincrease 0)

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

```
ts_o_clokod_#files=8
ts_o_clokod_size=1263m
ts_o_clokod_storage=(initial 8k next 25m maxextents unlimited pctincrease 0)

ts_o_op_#files=8
ts_o_op_size=1263m
ts_o_op_storage=(initial 8k next 25m maxextents unlimited pctincrease 0)

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

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

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

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

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

### loading  parameters

load_type = delim
load_field_terminator = '|'
load_tables = lineitem,orders,partsupp,parts,customer,supplier,nation,region
load_flatfile_area = o:\tpcd\data\hundgig\

load_controlfile_area = k:\tpcd\ctl\
load_otherfile_area = k:\tpcd\log\hundgig\
load_insert_type = append
load_deg_parallel = 8

# table load definitions

tab_nation_load_degpar = 1
tab_region_load_degpar = 1

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

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

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

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

### loading storage parameters

tab_lineitem_storage = (initial 8k next 1899m freelists 8 freelist groups 9 pctincrease 0)
tab_orders_storage = (initial 8k next 1100m freelists 8 freelist groups 9 pctincrease 0)
tab_parts_storage = (initial 8k next 453m pctincrease 0)
tab_partsupp_storage = (initial 8k next 1803m pctincrease 0)
tab_customer_storage = (initial 8k next 453m pctincrease 0)
tab_supplier_storage = (initial 8k next 48m pctincrease 0)
tab_nation_loadextent = 8k
tab_region_loadextent = 8k

# table creation parameters

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

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

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

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

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

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

tab_nation_ts = ts_nation
tab_region_ts = ts_region

tab_nation_cache = true
tab_region_cache = true

ind_indices = l_ored,l_pqesod,o_op,ps_pksk,o_clokod,ps_spsa,s_skey,r_rn
ind_unrecoverable=true
ind_pardeg=16

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

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

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

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

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

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

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

ind_r_rn_table = region
ind_r_rn_tabcols = r_name,r_regionkey
ind_r_rn_ts = ts_r_rn
ind_r_rn_pardeg=1
ind_r_rn_unique=1

### Parameters for analyzing objects

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

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

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

anl_l_ored_type = index
anl_l_ored_estimate = sample 200000 rows
anl_o_op_type = index
anl_o_op_estimate = sample 200000 rows
anl_l_pqesod_type = index
anl_l_pqesod_estimate = sample 200000 rows
anl_o_clokod_type = index
anl_o_clokod_estimate = sample 200000 rows
anl_ps_pksk_type = index
anl_ps_pksk_estimate = sample 200000 rows
anl_s_skey_type = index
anl_s_skey_estimate = sample 2000 rows
anl_ps_spsa_type = index
anl_ps_spsa_estimate = sample 200000 rows
anl_r_rn = index
anl_r_rn_estimate = sample 200000 rows

# Analyze tables

anl_lineitem_type = table
anl_lineitem_estimate = sample 200000 rows
anl_orders_type = table
anl_orders_estimate = sample 200000 rows
anl_partsupp_type = table
anl_partsupp_estimate = sample 200000 rows
anl_parts_type = table
anl_parts_estimate = sample 200000 rows
anl_customer_type = table
anl_customer_estimate = sample 200000 rows
anl_supplier_type = table
anl_supplier_estimate = sample 2000 rows
anl_nation_type = table
anl_region_type = table

# Note: Analyze historgram is not shown in the DIGITAL FDR

anl_lineitem_h_type = table
anl_lineitem_h_estimate = l_orderkey size 250, l_shipdate_size 250,&
 l_commitdate size 250, l_receiptdate size 250, l_suppkey size 250,&
 l_partkey size 250 sample 200000 rows
anl_orders_h_type = table
anl_orders_h_estimate = o_orderkey size 250, o_orderdate size 250,&
 o_clerk size 250, o_custkey size 250 sample 200000 rows
anl_partsupp_h_type = table
anl_partsupp_h_estimate = ps_suppkey size 250, ps_partkey size 250&
 sample 200000 rows
anl_parts_h_type = table
anl_parts_h_estimate = p_name size 250, p_brand size 250,&
 p_container size 250, p_type size 250, p_size 50 sample 200000 rows
anl_region_h_type = table
anl_region_h_compute =r_name size 5, r_regionkey size 5
anl_nation_h_type = table
anl_nation_h_compute =n_name size 25, n_nationkey size 25

### Parameters for explaining and querying

qry_all_queries=q1,q4b,q15b,q10,q11,q6,q2,q16,q14c,q8b,q12b,q17b,q3,q5,q13,q7,q9
qry_explain_queries=allqueries
qry_run_queries=q1,q4b,q15b,q10,q11,q6,q2,q16,q14c,q8b,q12b,q17b,q3,q5,q13,q7,q9
qry_getplan_area=d:\tpcd\getplan\
qry_out_area=d:\tpcd\out\
qry_config=20_1
qry_separate_queries=false
alter_tables=lineitem,orders,customer,parts,partsupp,supplier
```

BUILD.DAT

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

%b-preproc

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

*load
\sqlldr73 {}

%e-preproc
%b-dbcre
*bgon=20
#############################################################################
###
# Database Creation Phase
*sql
{
shutdown abort
}
*wait
# creating database and initial rollback segment
*sql
{
startup pfile=c:\tpcd\dbs\hundgig\p_build.ora nomount
create database
  controlfile reuse
  logfile '\\.\PHYSICALDRIVE9' size 1900m reuse,
```

```
        '\\.\PHYSICALDRIVE10' size 1900m reuse
  datafile 'c:\tpcd\dbs\hundgig\sys.dbf' size 499m reuse
  maxdatafiles 1020
;

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

alter rollback segment t_rs1 online;

shutdown
}
*wait
*sql
{
startup pfile=c:\tpcd\dbs\hundgig\p_build.ora
}
*wait
# creating extra rollback segments
*sql
{
create public rollback segment r1 storage (initial 200k next 200k);
create public rollback segment r2 storage (initial 200k next 200k);
create public rollback segment r3 storage (initial 200k next 200k);
create public rollback segment r4 storage (initial 200k next 200k);
create public rollback segment r5 storage (initial 200k next 200k);
create public rollback segment r6 storage (initial 200k next 200k);
create public rollback segment r7 storage (initial 200k next 200k);
create public rollback segment r8 storage (initial 200k next 200k);
create public rollback segment r9 storage (initial 200k next 200k);
create public rollback segment r10 storage (initial 200k next 200k);
create public rollback segment r11 storage (initial 200k next 200k);
create public rollback segment r12 storage (initial 200k next 200k);
create public rollback segment r13 storage (initial 200k next 200k);
create public rollback segment r14 storage (initial 200k next 200k);
create public rollback segment r15 storage (initial 200k next 200k);
create public rollback segment r16 storage (initial 200k next 200k);
create public rollback segment r17 storage (initial 200k next 200k);
create public rollback segment r18 storage (initial 200k next 200k);
create public rollback segment r19 storage (initial 200k next 200k);
create public rollback segment r20 storage (initial 200k next 200k);
create public rollback segment r21 storage (initial 200k next 200k);
create public rollback segment r22 storage (initial 200k next 200k);
create public rollback segment r23 storage (initial 200k next 200k);
create public rollback segment r24 storage (initial 200k next 200k);
create public rollback segment r25 storage (initial 200k next 200k);
create public rollback segment r26 storage (initial 200k next 200k);
create public rollback segment r27 storage (initial 200k next 200k);
create public rollback segment r28 storage (initial 200k next 200k);
create public rollback segment r29 storage (initial 200k next 200k);
create public rollback segment r30 storage (initial 200k next 200k);
}
*wait
# creating extra logfile threads
# building data dictionary
*sql
{
set termout off
set echo off
@k:\orant\rdbms73\admin\catalog.sql
@k:\orant\rdbms73\admin\catparr.sql
@k:\orant\rdbms73\admin\catproc.sql
@k:\orant\rdbms73\admin\utlxplan.sql
}
*wait
*sql
{
shutdown
}
*wait
*sql
{
startup pfile=c:\tpcd\dbs\hundgig\p_build.ora
}
*wait
*wait
*bgoff
%e-dbcre
%b-sccre
*bgon=20
#############################################################################
###
# Schema Creation Phase
# creating tpcd user
*sql
{
drop user tpcd cascade;
grant resource,unlimited tablespace,connect
  to tpcd identified by tpcd;
}
*wait
# creating data tablespaces, datafiles, and tables
# creating tpcd's ts_lineitem tablespace
*sql
{
drop tablespace ts_lineitem including contents;
create tablespace ts_lineitem
  datafile 'c:\tpcd\dbs\hundgig\ts_lineitem1.dbf' size 9500m reuse
  default storage (initial 8k next 1899m pctincrease 0)
;
}
# creating tpcd's ts_orders tablespace
*sql
{
drop tablespace ts_orders including contents;
create tablespace ts_orders
  datafile 'c:\tpcd\dbs\hundgig\ts_orders1.dbf' size 2350m reuse
  default storage (initial 8k next 1100m pctincrease 0)
;
}
# creating tpcd's ts_parts tablespace
*sql
{
drop tablespace ts_parts including contents;
create tablespace ts_parts
  datafile 'c:\tpcd\dbs\hundgig\ts_parts1.dbf' size 455m reuse
  default storage (initial 8k next 453m pctincrease 0)
;
}
# creating tpcd's ts_partsupp tablespace
*sql
{
drop tablespace ts_partsupp including contents;
create tablespace ts_partsupp
  datafile 'c:\tpcd\dbs\hundgig\ts_partsupp1.dbf' size 1805m reuse
  default storage (initial 8k next 1803m pctincrease 0)
;
}
# creating tpcd's ts_customer tablespace
*sql
{
drop tablespace ts_customer including contents;
create tablespace ts_customer
  datafile 'c:\tpcd\dbs\hundgig\ts_customer1.dbf' size 455m reuse
  default storage (initial 8k next 453m pctincrease 0)
;
}
# creating tpcd's ts_supplier tablespace
*sql
{
drop tablespace ts_supplier including contents;
create tablespace ts_supplier
  datafile 'c:\tpcd\dbs\hundgig\ts_supplier1.dbf' size 50m reuse
  default storage (initial 8k next 48m pctincrease 0)
;
}
# creating tpcd's ts_nation tablespace
*sql
{
drop tablespace ts_nation including contents;
create tablespace ts_nation
  datafile 'd:\tpcd\dbs\hundgig\nation.dbf' size 17k reuse
  default storage (initial 8k next 8k pctincrease 0)
;
}
# creating tpcd's ts_region tablespace
*sql
{
drop tablespace ts_region including contents;
create tablespace ts_region
  datafile 'e:\tpcd\dbs\hundgig\region.dbf' size 17k reuse
  default storage (initial 8k next 8k pctincrease 0)
;
}
# creating tpcd's ts_l_ored tablespace
```

```
*sql
{
drop tablespace ts_l_ored including contents;
create tablespace ts_l_ored
  datafile 'c:\tpcd\dbs\hundgig\ts_l_ored1.dbf' size 3171m reuse
  default storage (initial 8k next 100m maxextents unlimited pctincrease 0)
;
}
# creating tpcd's ts_l_pqesod tablespace
*sql
{
drop tablespace ts_l_pqesod including contents;
create tablespace ts_l_pqesod
  datafile 'c:\tpcd\dbs\hundgig\ts_l_pqesod1.dbf' size 3785m reuse
  default storage (initial 8k next 100m maxextents unlimited pctincrease 0)
;
}
# creating tpcd's ts_o_op tablespace
*sql
{
drop tablespace ts_o_op including contents;
create tablespace ts_o_op
  datafile 'c:\tpcd\dbs\hundgig\ts_o_op1.dbf' size 1263m reuse
  default storage (initial 8k next 25m maxextents unlimited pctincrease 0)
;
}
# creating tpcd's ts_ps_pksk tablespace
*sql
{
drop tablespace ts_ps_pksk including contents;
create tablespace ts_ps_pksk
  datafile 'c:\tpcd\dbs\hundgig\ts_ps_pksk1.dbf' size 948m reuse
  default storage (initial 8k next 20m maxextents unlimited pctincrease 0)
;
}
# creating tpcd's ts_o_clokod tablespace
*sql
{
drop tablespace ts_o_clokod including contents;
create tablespace ts_o_clokod
  datafile 'c:\tpcd\dbs\hundgig\ts_o_clokod1.dbf' size 1263m reuse
  default storage (initial 8k next 25m maxextents unlimited pctincrease 0)
;
}
# creating tpcd's ts_ps_spsa tablespace
*sql
{
drop tablespace ts_ps_spsa including contents;
create tablespace ts_ps_spsa
  datafile 'c:\tpcd\dbs\hundgig\ts_ps_spsa1.dbf' size 1128m reuse
  default storage (initial 8k next 20m maxextents unlimited pctincrease 0)
;
}
# creating tpcd's ts_s_skey tablespace
*sql
{
drop tablespace ts_s_skey including contents;
create tablespace ts_s_skey
  datafile 'c:\tpcd\dbs\hundgig\ts_s_skey1.dbf' size 103m reuse
  default storage (initial 8k next 20m maxextents unlimited pctincrease 0)
;
}
# creating tpcd's ts_r_rn tablespace
*sql
{
drop tablespace ts_r_rn including contents;
create tablespace ts_r_rn
  datafile 'c:\tpcd\dbs\hundgig\ts_r_rn1.dbf' size 17k reuse
  default storage (initial 8k next 8k maxextents unlimited pctincrease 0)
;
}
# creating tpcd's ts_temp tablespace
*sql
{
drop tablespace ts_temp including contents;
create tablespace ts_temp temporary
  datafile 'c:\tpcd\dbs\hundgig\ts_temp1.dbf' size 9900m reuse
  default storage (initial 8k next 500m maxextents unlimited pctincrease 0)
;
}
*wait
*sql
{
```

```
connect tpcd/tpcd;
set timing on;
drop table lineitem;
create table lineitem (
    l_shipdate          date ,
    l_orderkey          number ,
    l_discount          number ,
    l_extendedprice     number ,
    l_suppkey           number ,
    l_quantity          number ,
    l_returnflag        char(1) ,
    l_partkey           number ,
    l_linestatus        char(1) ,
    l_tax               number ,
    l_commitdate        date ,
    l_receiptdate       date ,
    l_shipmode          varchar(10) ,
    l_linenumber        number ,
    l_shipinstruct      varchar(25) ,
    l_comment           varchar(44)
)
pctfree 1
pctused 99
initrans 10
tablespace ts_lineitem
storage (initial 8k next 1899m freelists 8 freelist groups 9 pctincrease 0)
parallel (degree 60 )
;
drop table orders;
create table orders (
    o_orderdate         date ,
    o_orderkey          number NOT NULL,
    o_custkey           number NOT NULL,
    o_orderpriority     varchar(15) ,
    o_shippriority      number ,
    o_clerk             varchar(15) ,
    o_orderstatus       char(1) ,
    o_totalprice        number ,
    o_comment           varchar(79)
)
pctfree 1
pctused 99
initrans 10
tablespace ts_orders
storage (initial 8k next 1100m freelists 8 freelist groups 9 pctincrease 0)
parallel (degree 60 )
;
drop table partsupp;
create table partsupp (
    ps_partkey          number NOT NULL,
    ps_suppkey          number NOT NULL,
    ps_supplycost       number NOT NULL,
    ps_availqty         number ,
    ps_comment          varchar(199)
)
pctfree 0
pctused 99
tablespace ts_partsupp
storage (initial 8k next 1803m pctincrease 0)
parallel (degree 60 )
;
drop table parts;
create table parts (
    p_partkey           number NOT NULL,
    p_type              varchar(25) ,
    p_size              number ,
    p_brand             varchar(10) ,
    p_name              varchar(55) ,
    p_container         varchar(10) ,
    p_mfgr              varchar(25) ,
    p_retailprice       number ,
    p_comment           varchar(23)
)
pctfree 0
pctused 99
tablespace ts_parts
storage (initial 8k next 453m pctincrease 0)
parallel (degree 60 )
;
drop table customer;
create table customer (
    c_custkey           number NOT NULL,
    c_mktsegment        varchar(10) ,
```

```
    c_nationkey         number ,
    c_name              varchar(25) ,
    c_address           varchar(40) ,
    c_phone             varchar(15) ,
    c_acctbal           number ,
    c_comment           varchar(117)
)
pctfree 0
pctused 99
tablespace ts_customer
storage (initial 8k next 453m pctincrease 0)
parallel (degree 60 )
;
drop table supplier;
create table supplier (
    s_suppkey           number NOT NULL,
    s_nationkey         number ,
    s_comment           varchar(101) ,
    s_name              varchar(25) ,
    s_address           varchar(40) ,
    s_phone             varchar(15) ,
    s_acctbal           number
)
pctfree 0
pctused 99
tablespace ts_supplier
storage (initial 8k next 48m pctincrease 0)
parallel (degree 60 )
;
drop table nation;
create table nation (
    n_nationkey         number NOT NULL,
    n_name              varchar(25) ,
    n_regionkey         number ,
    n_comment           varchar(152)
)
tablespace ts_nation
cache
;
drop table region;
create table region (
    r_regionkey         number NOT NULL,
    r_name              varchar(25) ,
    r_comment           varchar(152)
)
tablespace ts_region
cache
;
}
*wait
# adding tpcd's ts_lineitem datafiles
*sql
{
alter tablespace ts_lineitem
  add datafile 'd:\tpcd\dbs\hundgig\ts_lineitem2.dbf' size 9500m reuse;
}
*sql
{
alter tablespace ts_lineitem
  add datafile 'e:\tpcd\dbs\hundgig\ts_lineitem3.dbf' size 9500m reuse;
}
*sql
{
alter tablespace ts_lineitem
  add datafile 'f:\tpcd\dbs\hundgig\ts_lineitem4.dbf' size 9500m reuse;
}
*sql
{
alter tablespace ts_lineitem
  add datafile 'g:\tpcd\dbs\hundgig\ts_lineitem5.dbf' size 9500m reuse;
}
*sql
{
alter tablespace ts_lineitem
  add datafile 'h:\tpcd\dbs\hundgig\ts_lineitem6.dbf' size 9500m reuse;
}
*sql
{
alter tablespace ts_lineitem
  add datafile 'i:\tpcd\dbs\hundgig\ts_lineitem7.dbf' size 9500m reuse;
}
*sql
{
alter tablespace ts_lineitem
  add datafile 'j:\tpcd\dbs\hundgig\ts_lineitem8.dbf' size 9500m reuse;
}
# adding tpcd's ts_orders datafiles
*sql
{
alter tablespace ts_orders
  add datafile 'd:\tpcd\dbs\hundgig\ts_orders2.dbf' size 2350m reuse;
}
*sql
{
alter tablespace ts_orders
  add datafile 'e:\tpcd\dbs\hundgig\ts_orders3.dbf' size 2350m reuse;
}
*sql
{
alter tablespace ts_orders
  add datafile 'f:\tpcd\dbs\hundgig\ts_orders4.dbf' size 2350m reuse;
}
*sql
{
alter tablespace ts_orders
  add datafile 'g:\tpcd\dbs\hundgig\ts_orders5.dbf' size 2350m reuse;
}
*sql
{
alter tablespace ts_orders
  add datafile 'h:\tpcd\dbs\hundgig\ts_orders6.dbf' size 2350m reuse;
}
*sql
{
alter tablespace ts_orders
  add datafile 'i:\tpcd\dbs\hundgig\ts_orders7.dbf' size 2350m reuse;
}
*sql
{
alter tablespace ts_orders
  add datafile 'j:\tpcd\dbs\hundgig\ts_orders8.dbf' size 2350m reuse;
}
# adding tpcd's ts_parts datafiles
*sql
{
alter tablespace ts_parts
  add datafile 'd:\tpcd\dbs\hundgig\ts_parts2.dbf' size 455m reuse;
}
*sql
{
alter tablespace ts_parts
  add datafile 'e:\tpcd\dbs\hundgig\ts_parts3.dbf' size 455m reuse;
}
*sql
{
alter tablespace ts_parts
  add datafile 'f:\tpcd\dbs\hundgig\ts_parts4.dbf' size 455m reuse;
}
*sql
{
alter tablespace ts_parts
  add datafile 'g:\tpcd\dbs\hundgig\ts_parts5.dbf' size 455m reuse;
}
*sql
{
alter tablespace ts_parts
  add datafile 'h:\tpcd\dbs\hundgig\ts_parts6.dbf' size 455m reuse;
}
*sql
{
alter tablespace ts_parts
  add datafile 'i:\tpcd\dbs\hundgig\ts_parts7.dbf' size 455m reuse;
}
*sql
{
alter tablespace ts_parts
  add datafile 'j:\tpcd\dbs\hundgig\ts_parts8.dbf' size 455m reuse;
}
# adding tpcd's ts_partsupp datafiles
*sql
{
alter tablespace ts_partsupp
  add datafile 'd:\tpcd\dbs\hundgig\ts_partsupp2.dbf' size 1805m reuse;
}
```

```
*sql
{
alter tablespace ts_partsupp
  add datafile 'e:\tpcd\dbs\hundgig\ts_partsupp3.dbf' size 1805m reuse;
}
*sql
{
alter tablespace ts_partsupp
  add datafile 'f:\tpcd\dbs\hundgig\ts_partsupp4.dbf' size 1805m reuse;
}
*sql
{
alter tablespace ts_partsupp
  add datafile 'g:\tpcd\dbs\hundgig\ts_partsupp5.dbf' size 1805m reuse;
}
*sql
{
alter tablespace ts_partsupp
  add datafile 'h:\tpcd\dbs\hundgig\ts_partsupp6.dbf' size 1805m reuse;
}
*sql
{
alter tablespace ts_partsupp
  add datafile 'i:\tpcd\dbs\hundgig\ts_partsupp7.dbf' size 1805m reuse;
}
*sql
{
alter tablespace ts_partsupp
  add datafile 'j:\tpcd\dbs\hundgig\ts_partsupp8.dbf' size 1805m reuse;
}
# adding tpcd's ts_customer datafiles
*sql
{
alter tablespace ts_customer
  add datafile 'd:\tpcd\dbs\hundgig\ts_customer2.dbf' size 455m reuse;
}
*sql
{
alter tablespace ts_customer
  add datafile 'e:\tpcd\dbs\hundgig\ts_customer3.dbf' size 455m reuse;
}
*sql
{
alter tablespace ts_customer
  add datafile 'f:\tpcd\dbs\hundgig\ts_customer4.dbf' size 455m reuse;
}
*sql
{
alter tablespace ts_customer
  add datafile 'g:\tpcd\dbs\hundgig\ts_customer5.dbf' size 455m reuse;
}
*sql
{
alter tablespace ts_customer
  add datafile 'h:\tpcd\dbs\hundgig\ts_customer6.dbf' size 455m reuse;
}
*sql
{
alter tablespace ts_customer
  add datafile 'i:\tpcd\dbs\hundgig\ts_customer7.dbf' size 455m reuse;
}
*sql
{
alter tablespace ts_customer
  add datafile 'j:\tpcd\dbs\hundgig\ts_customer8.dbf' size 455m reuse;
}
# adding tpcd's ts_supplier datafiles
*sql
{
alter tablespace ts_supplier
  add datafile 'd:\tpcd\dbs\hundgig\ts_supplier2.dbf' size 50m reuse;
}
*sql
{
alter tablespace ts_supplier
  add datafile 'e:\tpcd\dbs\hundgig\ts_supplier3.dbf' size 50m reuse;
}
*sql
{
alter tablespace ts_supplier
  add datafile 'f:\tpcd\dbs\hundgig\ts_supplier4.dbf' size 50m reuse;
}
*sql
{
alter tablespace ts_supplier
  add datafile 'g:\tpcd\dbs\hundgig\ts_supplier5.dbf' size 50m reuse;
}
*sql
{
alter tablespace ts_supplier
  add datafile 'h:\tpcd\dbs\hundgig\ts_supplier6.dbf' size 50m reuse;
}
*sql
{
alter tablespace ts_supplier
  add datafile 'i:\tpcd\dbs\hundgig\ts_supplier7.dbf' size 50m reuse;
}
*sql
{
alter tablespace ts_supplier
  add datafile 'j:\tpcd\dbs\hundgig\ts_supplier8.dbf' size 50m reuse;
}
# adding tpcd's ts_l_ored datafiles
*sql
{
alter tablespace ts_l_ored
  add datafile 'd:\tpcd\dbs\hundgig\ts_l_ored2.dbf' size 3171m reuse;
}
*sql
{
alter tablespace ts_l_ored
  add datafile 'e:\tpcd\dbs\hundgig\ts_l_ored3.dbf' size 3171m reuse;
}
*sql
{
alter tablespace ts_l_ored
  add datafile 'f:\tpcd\dbs\hundgig\ts_l_ored4.dbf' size 3171m reuse;
}
*sql
{
alter tablespace ts_l_ored
  add datafile 'g:\tpcd\dbs\hundgig\ts_l_ored5.dbf' size 3171m reuse;
}
*sql
{
alter tablespace ts_l_ored
  add datafile 'h:\tpcd\dbs\hundgig\ts_l_ored6.dbf' size 3171m reuse;
}
*sql
{
alter tablespace ts_l_ored
  add datafile 'i:\tpcd\dbs\hundgig\ts_l_ored7.dbf' size 3171m reuse;
}
*sql
{
alter tablespace ts_l_ored
  add datafile 'j:\tpcd\dbs\hundgig\ts_l_ored8.dbf' size 3171m reuse;
}
# adding tpcd's ts_l_pqesod datafiles
*sql
{
alter tablespace ts_l_pqesod
  add datafile 'd:\tpcd\dbs\hundgig\ts_l_pqesod2.dbf' size 3785m reuse;
}
*sql
{
alter tablespace ts_l_pqesod
  add datafile 'e:\tpcd\dbs\hundgig\ts_l_pqesod3.dbf' size 3785m reuse;
}
*sql
{
alter tablespace ts_l_pqesod
  add datafile 'f:\tpcd\dbs\hundgig\ts_l_pqesod4.dbf' size 3785m reuse;
}
*sql
{
alter tablespace ts_l_pqesod
  add datafile 'g:\tpcd\dbs\hundgig\ts_l_pqesod5.dbf' size 3785m reuse;
}
*sql
{
alter tablespace ts_l_pqesod
  add datafile 'h:\tpcd\dbs\hundgig\ts_l_pqesod6.dbf' size 3785m reuse;
}
*sql
{
```

```
alter tablespace ts_l_pqesod
  add datafile 'i:\tpcd\dbs\hundgig\ts_l_pqesod7.dbf' size 3785m reuse;
}
*sql
{
alter tablespace ts_l_pqesod
  add datafile 'j:\tpcd\dbs\hundgig\ts_l_pqesod8.dbf' size 3785m reuse;
}
# adding tpcd's ts_o_op datafiles
*sql
{
alter tablespace ts_o_op
  add datafile 'd:\tpcd\dbs\hundgig\ts_o_op2.dbf' size 1263m reuse;
}
*sql
{
alter tablespace ts_o_op
  add datafile 'e:\tpcd\dbs\hundgig\ts_o_op3.dbf' size 1263m reuse;
}
*sql
{
alter tablespace ts_o_op
  add datafile 'f:\tpcd\dbs\hundgig\ts_o_op4.dbf' size 1263m reuse;
}
*sql
{
alter tablespace ts_o_op
  add datafile 'g:\tpcd\dbs\hundgig\ts_o_op5.dbf' size 1263m reuse;
}
*sql
{
alter tablespace ts_o_op
  add datafile 'h:\tpcd\dbs\hundgig\ts_o_op6.dbf' size 1263m reuse;
}
*sql
{
alter tablespace ts_o_op
  add datafile 'i:\tpcd\dbs\hundgig\ts_o_op7.dbf' size 1263m reuse;
}
*sql
{
alter tablespace ts_o_op
  add datafile 'j:\tpcd\dbs\hundgig\ts_o_op8.dbf' size 1263m reuse;
}
# adding tpcd's ts_ps_pksk datafiles
*sql
{
alter tablespace ts_ps_pksk
  add datafile 'd:\tpcd\dbs\hundgig\ts_ps_pksk2.dbf' size 948m reuse;
}
*sql
{
alter tablespace ts_ps_pksk
  add datafile 'e:\tpcd\dbs\hundgig\ts_ps_pksk3.dbf' size 948m reuse;
}
*sql
{
alter tablespace ts_ps_pksk
  add datafile 'f:\tpcd\dbs\hundgig\ts_ps_pksk4.dbf' size 948m reuse;
}
*sql
{
alter tablespace ts_ps_pksk
  add datafile 'g:\tpcd\dbs\hundgig\ts_ps_pksk5.dbf' size 948m reuse;
}
*sql
{
alter tablespace ts_ps_pksk
  add datafile 'h:\tpcd\dbs\hundgig\ts_ps_pksk6.dbf' size 948m reuse;
}
*sql
{
alter tablespace ts_ps_pksk
  add datafile 'i:\tpcd\dbs\hundgig\ts_ps_pksk7.dbf' size 948m reuse;
}
*sql
{
alter tablespace ts_ps_pksk
  add datafile 'j:\tpcd\dbs\hundgig\ts_ps_pksk8.dbf' size 948m reuse;
}
# adding tpcd's ts_o_clokod datafiles
*sql
{
alter tablespace ts_o_clokod
  add datafile 'd:\tpcd\dbs\hundgig\ts_o_clokod2.dbf' size 1263m reuse;
}
*sql
{
alter tablespace ts_o_clokod
  add datafile 'e:\tpcd\dbs\hundgig\ts_o_clokod3.dbf' size 1263m reuse;
}
*sql
{
alter tablespace ts_o_clokod
  add datafile 'f:\tpcd\dbs\hundgig\ts_o_clokod4.dbf' size 1263m reuse;
}
*sql
{
alter tablespace ts_o_clokod
  add datafile 'g:\tpcd\dbs\hundgig\ts_o_clokod5.dbf' size 1263m reuse;
}
*sql
{
alter tablespace ts_o_clokod
  add datafile 'h:\tpcd\dbs\hundgig\ts_o_clokod6.dbf' size 1263m reuse;
}
*sql
{
alter tablespace ts_o_clokod
  add datafile 'i:\tpcd\dbs\hundgig\ts_o_clokod7.dbf' size 1263m reuse;
}
*sql
{
alter tablespace ts_o_clokod
  add datafile 'j:\tpcd\dbs\hundgig\ts_o_clokod8.dbf' size 1263m reuse;
}
# adding tpcd's ts_ps_spsa datafiles
*sql
{
alter tablespace ts_ps_spsa
  add datafile 'd:\tpcd\dbs\hundgig\ts_ps_spsa2.dbf' size 1128m reuse;
}
*sql
{
alter tablespace ts_ps_spsa
  add datafile 'e:\tpcd\dbs\hundgig\ts_ps_spsa3.dbf' size 1128m reuse;
}
*sql
{
alter tablespace ts_ps_spsa
  add datafile 'f:\tpcd\dbs\hundgig\ts_ps_spsa4.dbf' size 1128m reuse;
}
*sql
{
alter tablespace ts_ps_spsa
  add datafile 'g:\tpcd\dbs\hundgig\ts_ps_spsa5.dbf' size 1128m reuse;
}
*sql
{
alter tablespace ts_ps_spsa
  add datafile 'h:\tpcd\dbs\hundgig\ts_ps_spsa6.dbf' size 1128m reuse;
}
*sql
{
alter tablespace ts_ps_spsa
  add datafile 'i:\tpcd\dbs\hundgig\ts_ps_spsa7.dbf' size 1128m reuse;
}
*sql
{
alter tablespace ts_ps_spsa
  add datafile 'j:\tpcd\dbs\hundgig\ts_ps_spsa8.dbf' size 1128m reuse;
}
# adding tpcd's ts_s_skey datafiles
*sql
{
alter tablespace ts_s_skey
  add datafile 'd:\tpcd\dbs\hundgig\ts_s_skey2.dbf' size 103m reuse;
}
*sql
{
alter tablespace ts_s_skey
  add datafile 'e:\tpcd\dbs\hundgig\ts_s_skey3.dbf' size 103m reuse;
}
*sql
```

```
{
alter tablespace ts_s_skey
  add datafile 'f:\tpcd\dbs\hundgig\ts_s_skey4.dbf' size 103m reuse;
}
*sql
{
alter tablespace ts_s_skey
  add datafile 'g:\tpcd\dbs\hundgig\ts_s_skey5.dbf' size 103m reuse;
}
*sql
{
alter tablespace ts_s_skey
  add datafile 'h:\tpcd\dbs\hundgig\ts_s_skey6.dbf' size 103m reuse;
}
*sql
{
alter tablespace ts_s_skey
  add datafile 'i:\tpcd\dbs\hundgig\ts_s_skey7.dbf' size 103m reuse;
}
*sql
{
alter tablespace ts_s_skey
  add datafile 'j:\tpcd\dbs\hundgig\ts_s_skey8.dbf' size 103m reuse;
}
# adding tpcd's ts_temp datafiles
*sql
{
alter tablespace ts_temp
  add datafile 'd:\tpcd\dbs\hundgig\ts_temp2.dbf' size 9900m reuse;
}
*sql
{
alter tablespace ts_temp
  add datafile 'e:\tpcd\dbs\hundgig\ts_temp3.dbf' size 9900m reuse;
}
*sql
{
alter tablespace ts_temp
  add datafile 'f:\tpcd\dbs\hundgig\ts_temp4.dbf' size 9900m reuse;
}
*sql
{
alter tablespace ts_temp
  add datafile 'g:\tpcd\dbs\hundgig\ts_temp5.dbf' size 9900m reuse;
}
*sql
{
alter tablespace ts_temp
  add datafile 'h:\tpcd\dbs\hundgig\ts_temp6.dbf' size 9900m reuse;
}
*sql
{
alter tablespace ts_temp
  add datafile 'i:\tpcd\dbs\hundgig\ts_temp7.dbf' size 9900m reuse;
}
*sql
{
alter tablespace ts_temp
  add datafile 'j:\tpcd\dbs\hundgig\ts_temp8.dbf' size 9900m reuse;
}
*wait
# altering tpcd's temporary tablespace
*sql
{
alter user tpcd temporary tablespace ts_temp;
}
*wait
*wait
*bgoff
%e-sccre
```

LOAD.DAT

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

%b-preproc

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

*load
\sqlldr73 {}

%e-preproc
%b-shutd
#############################################################################
###
# Shutdown Database - All Instances
*time=Shutting down database
*wait
*sql
{
shutdown
}
*wait
*time=Database shutdown
*wait
%e-shutd
%b-start
#############################################################################
###
# Startup Database - All Instances
*time=Startup database
*wait
*sql
{
startup pfile=c:\tpcd\dbs\hundgig\p_run.ora
}
*wait
*time=Database started up
*wait
%e-start
%b-scuto
*bgoff
#############################################################################
###
# Schema Creation Phase - User and Tables ONLY (no datafiles)
*time=Begin TPCD table cleanup
*wait
*sql
{
connect tpcd/tpcd;
drop index l_ored;
drop index l_pqesod;
drop index o_op;
drop index ps_pksk;
drop index o_clokod;
drop index ps_spsa;
drop index s_skey;
drop index r_rn;
}
*wait
*sql
{
connect tpcd/tpcd;
drop table lineitem;
drop table orders;
drop table partsupp;
drop table parts;
drop table customer;
drop table supplier;
drop table nation;
drop table region;
}
*wait
*sql
{
connect internal;
alter tablespace ts_lineitem coalesce;
alter tablespace ts_orders coalesce;
alter tablespace ts_parts coalesce;
alter tablespace ts_partsupp coalesce;
alter tablespace ts_customer coalesce;
alter tablespace ts_supplier coalesce;
alter tablespace ts_nation coalesce;
alter tablespace ts_region coalesce;
alter tablespace ts_l_ored coalesce;
alter tablespace ts_l_pqesod coalesce;
alter tablespace ts_o_op coalesce;
alter tablespace ts_ps_pksk coalesce;
```

```
alter tablespace ts_o_clokod coalesce;
alter tablespace ts_ps_spsa coalesce;
alter tablespace ts_s_skey coalesce;
alter tablespace ts_r_rn coalesce;
alter tablespace ts_temp coalesce;
alter tablespace system coalesce;
connect sys/change_on_install;
@k:\tpcd\sql\orst_cre;
}
*wait
*sql
{
shutdown
}
*wait
*time=Restarting the database for preparation
*wait
*sql
{
startup pfile=c:\tpcd\dbs\hundgig\p_load.ora
}
*wait
*time=End TPCD table cleanup
*wait
*bgoff
*time=Begin TPCD database preparation
*wait
*time=Begin TPCD table creation
*sql
{
connect tpcd/tpcd;
set timing on;
drop table lineitem;
create table lineitem (
   l_shipdate          date ,
   l_orderkey          number ,
   l_discount          number ,
   l_extendedprice     number ,
   l_suppkey           number ,
   l_quantity          number ,
   l_returnflag        char(1) ,
   l_partkey           number ,
   l_linestatus        char(1) ,
   l_tax               number ,
   l_commitdate        date ,
   l_receiptdate       date ,
   l_shipmode          varchar(10) ,
   l_linenumber        number ,
   l_shipinstruct      varchar(25) ,
   l_comment           varchar(44)
)
pctfree 1
pctused 99
initrans 10
tablespace ts_lineitem
storage (initial 8k next 1899m freelists 8 freelist groups 9 pctincrease 0)
parallel (degree 60 )
;
drop table orders;
create table orders (
   o_orderdate         date ,
   o_orderkey          number NOT NULL,
   o_custkey           number NOT NULL,
   o_orderpriority     varchar(15) ,
   o_shippriority      number ,
   o_clerk             varchar(15) ,
   o_orderstatus       char(1) ,
   o_totalprice        number ,
   o_comment           varchar(79)
)
pctfree 1
pctused 99
initrans 10
tablespace ts_orders
storage (initial 8k next 1100m freelists 8 freelist groups 9 pctincrease 0)
parallel (degree 60 )
;
drop table partsupp;
create table partsupp (
   ps_partkey          number NOT NULL,
   ps_suppkey          number NOT NULL,
   ps_supplycost       number NOT NULL,
   ps_availqty         number ,
   ps_comment          varchar(199)
)
pctfree 0
pctused 99
tablespace ts_partsupp
storage (initial 8k next 1803m pctincrease 0)
parallel (degree 60 )
;
drop table parts;
create table parts (
   p_partkey           number NOT NULL,
   p_type              varchar(25) ,
   p_size              number ,
   p_brand             varchar(10) ,
   p_name              varchar(55) ,
   p_container         varchar(10) ,
   p_mfgr              varchar(25) ,
   p_retailprice       number ,
   p_comment           varchar(23)
)
pctfree 0
pctused 99
tablespace ts_parts
storage (initial 8k next 453m pctincrease 0)
parallel (degree 60 )
;
drop table customer;
create table customer (
   c_custkey           number NOT NULL,
   c_mktsegment        varchar(10) ,
   c_nationkey         number ,
   c_name              varchar(25) ,
   c_address           varchar(40) ,
   c_phone             varchar(15) ,
   c_acctbal           number ,
   c_comment           varchar(117)
)
pctfree 0
pctused 99
tablespace ts_customer
storage (initial 8k next 453m pctincrease 0)
parallel (degree 60 )
;
drop table supplier;
create table supplier (
   s_suppkey           number NOT NULL,
   s_nationkey         number ,
   s_comment           varchar(101) ,
   s_name              varchar(25) ,
   s_address           varchar(40) ,
   s_phone             varchar(15) ,
   s_acctbal           number
)
pctfree 0
pctused 99
tablespace ts_supplier
storage (initial 8k next 48m pctincrease 0)
parallel (degree 60 )
;
drop table nation;
create table nation (
   n_nationkey         number NOT NULL,
   n_name              varchar(25) ,
   n_regionkey         number ,
   n_comment           varchar(152)
)
tablespace ts_nation
cache
;
drop table region;
create table region (
   r_regionkey         number NOT NULL,
   r_name              varchar(25) ,
   r_comment           varchar(152)
)
tablespace ts_region
cache
;
}
*time=End TPCD table creation
*wait
```

```
*bgoff
*wait
*bgoff
%e-scuto
%b-dapop
*bgon=20
###############################################################################
###
# Database Population Phase
*wait
*time=Begin TPCD database load, begin lineitem
*wait
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem1.log
data=o:\tpcd\hundgig\data\lineitem.tbl.1
FILE=C:\TPCD\DBS\HUNDGIG\TS_LINEITEM1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem2.log
data=o:\tpcd\hundgig\data\lineitem.tbl.2
FILE=D:\TPCD\DBS\HUNDGIG\TS_LINEITEM2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem3.log
data=o:\tpcd\hundgig\data\lineitem.tbl.3
FILE=E:\TPCD\DBS\HUNDGIG\TS_LINEITEM3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem4.log
data=o:\tpcd\hundgig\data\lineitem.tbl.4
FILE=F:\TPCD\DBS\HUNDGIG\TS_LINEITEM4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem5.log
data=o:\tpcd\hundgig\data\lineitem.tbl.5
FILE=G:\TPCD\DBS\HUNDGIG\TS_LINEITEM5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem6.log
data=o:\tpcd\hundgig\data\lineitem.tbl.6
FILE=H:\TPCD\DBS\HUNDGIG\TS_LINEITEM6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem7.log
data=o:\tpcd\hundgig\data\lineitem.tbl.7
FILE=I:\TPCD\DBS\HUNDGIG\TS_LINEITEM7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem8.log
data=o:\tpcd\hundgig\data\lineitem.tbl.8
FILE=J:\TPCD\DBS\HUNDGIG\TS_LINEITEM8.DBF direct=true parallel=true
}
*wait
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem9.log
data=o:\tpcd\hundgig\data\lineitem.tbl.9
FILE=C:\TPCD\DBS\HUNDGIG\TS_LINEITEM1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem10.log
data=o:\tpcd\hundgig\data\lineitem.tbl.10
FILE=D:\TPCD\DBS\HUNDGIG\TS_LINEITEM2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem11.log
data=o:\tpcd\hundgig\data\lineitem.tbl.11
FILE=E:\TPCD\DBS\HUNDGIG\TS_LINEITEM3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem12.log
data=o:\tpcd\hundgig\data\lineitem.tbl.12
FILE=F:\TPCD\DBS\HUNDGIG\TS_LINEITEM4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem13.log
data=o:\tpcd\hundgig\data\lineitem.tbl.13
FILE=G:\TPCD\DBS\HUNDGIG\TS_LINEITEM5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem14.log
data=o:\tpcd\hundgig\data\lineitem.tbl.14
FILE=H:\TPCD\DBS\HUNDGIG\TS_LINEITEM6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem15.log
data=o:\tpcd\hundgig\data\lineitem.tbl.15
FILE=I:\TPCD\DBS\HUNDGIG\TS_LINEITEM7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem16.log
data=o:\tpcd\hundgig\data\lineitem.tbl.16
FILE=J:\TPCD\DBS\HUNDGIG\TS_LINEITEM8.DBF direct=true parallel=true
}
*wait
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem17.log
data=o:\tpcd\hundgig\data\lineitem.tbl.17
FILE=C:\TPCD\DBS\HUNDGIG\TS_LINEITEM1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem18.log
data=o:\tpcd\hundgig\data\lineitem.tbl.18
FILE=D:\TPCD\DBS\HUNDGIG\TS_LINEITEM2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem19.log
data=o:\tpcd\hundgig\data\lineitem.tbl.19
FILE=E:\TPCD\DBS\HUNDGIG\TS_LINEITEM3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem20.log
data=o:\tpcd\hundgig\data\lineitem.tbl.20
FILE=F:\TPCD\DBS\HUNDGIG\TS_LINEITEM4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem21.log
data=o:\tpcd\hundgig\data\lineitem.tbl.21
FILE=G:\TPCD\DBS\HUNDGIG\TS_LINEITEM5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem22.log
data=o:\tpcd\hundgig\data\lineitem.tbl.22
FILE=H:\TPCD\DBS\HUNDGIG\TS_LINEITEM6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem23.log
data=o:\tpcd\hundgig\data\lineitem.tbl.23
FILE=I:\TPCD\DBS\HUNDGIG\TS_LINEITEM7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem24.log
data=o:\tpcd\hundgig\data\lineitem.tbl.24
FILE=J:\TPCD\DBS\HUNDGIG\TS_LINEITEM8.DBF direct=true parallel=true
}
*wait
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem25.log
data=o:\tpcd\hundgig\data\lineitem.tbl.25
FILE=C:\TPCD\DBS\HUNDGIG\TS_LINEITEM1.DBF direct=true parallel=true
```

```
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem26.log
data=o:\tpcd\hundgig\data\lineitem.tbl.26
FILE=D:\TPCD\DBS\HUNDGIG\TS_LINEITEM2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem27.log
data=o:\tpcd\hundgig\data\lineitem.tbl.27
FILE=E:\TPCD\DBS\HUNDGIG\TS_LINEITEM3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem28.log
data=o:\tpcd\hundgig\data\lineitem.tbl.28
FILE=F:\TPCD\DBS\HUNDGIG\TS_LINEITEM4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem29.log
data=o:\tpcd\hundgig\data\lineitem.tbl.29
FILE=G:\TPCD\DBS\HUNDGIG\TS_LINEITEM5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem30.log
data=o:\tpcd\hundgig\data\lineitem.tbl.30
FILE=H:\TPCD\DBS\HUNDGIG\TS_LINEITEM6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem31.log
data=o:\tpcd\hundgig\data\lineitem.tbl.31
FILE=I:\TPCD\DBS\HUNDGIG\TS_LINEITEM7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem32.log
data=o:\tpcd\hundgig\data\lineitem.tbl.32
FILE=J:\TPCD\DBS\HUNDGIG\TS_LINEITEM8.DBF direct=true parallel=true
}
*wait
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem33.log
data=o:\tpcd\hundgig\data\lineitem.tbl.33
FILE=C:\TPCD\DBS\HUNDGIG\TS_LINEITEM1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem34.log
data=o:\tpcd\hundgig\data\lineitem.tbl.34
FILE=D:\TPCD\DBS\HUNDGIG\TS_LINEITEM2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem35.log
data=o:\tpcd\hundgig\data\lineitem.tbl.35
FILE=E:\TPCD\DBS\HUNDGIG\TS_LINEITEM3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem36.log
data=o:\tpcd\hundgig\data\lineitem.tbl.36
FILE=F:\TPCD\DBS\HUNDGIG\TS_LINEITEM4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem37.log
data=o:\tpcd\hundgig\data\lineitem.tbl.37
FILE=G:\TPCD\DBS\HUNDGIG\TS_LINEITEM5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem38.log
data=o:\tpcd\hundgig\data\lineitem.tbl.38
FILE=H:\TPCD\DBS\HUNDGIG\TS_LINEITEM6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem39.log
data=o:\tpcd\hundgig\data\lineitem.tbl.39
FILE=I:\TPCD\DBS\HUNDGIG\TS_LINEITEM7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\lineitem.ctl log=k:\tpcd\log\hundgig\lineitem40.log
data=o:\tpcd\hundgig\data\lineitem.tbl.40
FILE=J:\TPCD\DBS\HUNDGIG\TS_LINEITEM8.DBF direct=true parallel=true
}
*wait
*time=End lineitem, Begin orders
*wait
*load
{
tpcd/tpcd control=k:\tpcd\ctl\orders.ctl log=k:\tpcd\log\hundgig\orders1.log
data=o:\tpcd\hundgig\data\order.tbl.1
FILE=C:\TPCD\DBS\HUNDGIG\TS_ORDERS1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\orders.ctl log=k:\tpcd\log\hundgig\orders2.log
data=o:\tpcd\hundgig\data\order.tbl.2
FILE=D:\TPCD\DBS\HUNDGIG\TS_ORDERS2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\orders.ctl log=k:\tpcd\log\hundgig\orders3.log
data=o:\tpcd\hundgig\data\order.tbl.3
FILE=E:\TPCD\DBS\HUNDGIG\TS_ORDERS3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\orders.ctl log=k:\tpcd\log\hundgig\orders4.log
data=o:\tpcd\hundgig\data\order.tbl.4 FILE=F:\TPCD\DBS\HUNDGIG\TS_ORDERS4.DBF
direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\orders.ctl log=k:\tpcd\log\hundgig\orders5.log
data=o:\tpcd\hundgig\data\order.tbl.5
FILE=G:\TPCD\DBS\HUNDGIG\TS_ORDERS5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\orders.ctl log=k:\tpcd\log\hundgig\orders6.log
data=o:\tpcd\hundgig\data\order.tbl.6
FILE=H:\TPCD\DBS\HUNDGIG\TS_ORDERS6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\orders.ctl log=k:\tpcd\log\hundgig\orders7.log
data=o:\tpcd\hundgig\data\order.tbl.7 FILE=I:\TPCD\DBS\HUNDGIG\TS_ORDERS7.DBF
direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\orders.ctl log=k:\tpcd\log\hundgig\orders8.log
data=o:\tpcd\hundgig\data\order.tbl.8 FILE=J:\TPCD\DBS\HUNDGIG\TS_ORDERS8.DBF
direct=true parallel=true
}
*wait
*load
{
tpcd/tpcd control=k:\tpcd\ctl\orders.ctl log=k:\tpcd\log\hundgig\orders9.log
data=o:\tpcd\hundgig\data\order.tbl.9
FILE=C:\TPCD\DBS\HUNDGIG\TS_ORDERS1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\orders.ctl log=k:\tpcd\log\hundgig\orders10.log
data=o:\tpcd\hundgig\data\order.tbl.10
FILE=D:\TPCD\DBS\HUNDGIG\TS_ORDERS2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\orders.ctl log=k:\tpcd\log\hundgig\orders11.log
data=o:\tpcd\hundgig\data\order.tbl.11
FILE=E:\TPCD\DBS\HUNDGIG\TS_ORDERS3.DBF direct=true parallel=true
}
*load
{
```

```
tpcd/tpcd control=k:\tpcd\ctl\orders.ctl log=k:\tpcd\log\hundgig\orders12.log
data=o:\tpcd\hundgig\data\order.tbl.12
FILE=F:\TPCD\DBS\HUNDGIG\TS_ORDERS4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\orders.ctl log=k:\tpcd\log\hundgig\orders13.log
data=o:\tpcd\hundgig\data\order.tbl.13
FILE=G:\TPCD\DBS\HUNDGIG\TS_ORDERS5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\orders.ctl log=k:\tpcd\log\hundgig\orders14.log
data=o:\tpcd\hundgig\data\order.tbl.14
FILE=H:\TPCD\DBS\HUNDGIG\TS_ORDERS6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\orders.ctl log=k:\tpcd\log\hundgig\orders15.log
data=o:\tpcd\hundgig\data\order.tbl.15
FILE=I:\TPCD\DBS\HUNDGIG\TS_ORDERS7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\orders.ctl log=k:\tpcd\log\hundgig\orders16.log
data=o:\tpcd\hundgig\data\order.tbl.16
FILE=J:\TPCD\DBS\HUNDGIG\TS_ORDERS8.DBF direct=true parallel=true
}
*wait
*time=End orders, Begin partsupp
*wait
*load
{
tpcd/tpcd control=k:\tpcd\ctl\partsupp.ctl log=k:\tpcd\log\hundgig\partsupp1.log
data=o:\tpcd\hundgig\data\partsupp.tbl.1
FILE=C:\TPCD\DBS\HUNDGIG\TS_PARTSUPP1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\partsupp.ctl log=k:\tpcd\log\hundgig\partsupp2.log
data=o:\tpcd\hundgig\data\partsupp.tbl.2
FILE=D:\TPCD\DBS\HUNDGIG\TS_PARTSUPP2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\partsupp.ctl log=k:\tpcd\log\hundgig\partsupp3.log
data=o:\tpcd\hundgig\data\partsupp.tbl.3
FILE=E:\TPCD\DBS\HUNDGIG\TS_PARTSUPP3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\partsupp.ctl log=k:\tpcd\log\hundgig\partsupp4.log
data=o:\tpcd\hundgig\data\partsupp.tbl.4
FILE=F:\TPCD\DBS\HUNDGIG\TS_PARTSUPP4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\partsupp.ctl log=k:\tpcd\log\hundgig\partsupp5.log
data=o:\tpcd\hundgig\data\partsupp.tbl.5
FILE=G:\TPCD\DBS\HUNDGIG\TS_PARTSUPP5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\partsupp.ctl log=k:\tpcd\log\hundgig\partsupp6.log
data=o:\tpcd\hundgig\data\partsupp.tbl.6
FILE=H:\TPCD\DBS\HUNDGIG\TS_PARTSUPP6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\partsupp.ctl log=k:\tpcd\log\hundgig\partsupp7.log
data=o:\tpcd\hundgig\data\partsupp.tbl.7
FILE=I:\TPCD\DBS\HUNDGIG\TS_PARTSUPP7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\partsupp.ctl log=k:\tpcd\log\hundgig\partsupp8.log
data=o:\tpcd\hundgig\data\partsupp.tbl.8
FILE=J:\TPCD\DBS\HUNDGIG\TS_PARTSUPP8.DBF direct=true parallel=true
}
*wait
*time=End partsupp, Begin parts
*wait
*load
{
tpcd/tpcd control=k:\tpcd\ctl\parts.ctl log=k:\tpcd\log\hundgig\parts1.log
data=o:\tpcd\hundgig\data\part.tbl.1 FILE=C:\TPCD\DBS\HUNDGIG\TS_PARTS1.DBF
direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\parts.ctl log=k:\tpcd\log\hundgig\parts2.log
data=o:\tpcd\hundgig\data\part.tbl.2 FILE=D:\TPCD\DBS\HUNDGIG\TS_PARTS2.DBF
direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\parts.ctl log=k:\tpcd\log\hundgig\parts3.log
data=o:\tpcd\hundgig\data\part.tbl.3 FILE=E:\TPCD\DBS\HUNDGIG\TS_PARTS3.DBF
direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\parts.ctl log=k:\tpcd\log\hundgig\parts4.log
data=o:\tpcd\hundgig\data\part.tbl.4 FILE=F:\TPCD\DBS\HUNDGIG\TS_PARTS4.DBF
direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\parts.ctl log=k:\tpcd\log\hundgig\parts5.log
data=o:\tpcd\hundgig\data\part.tbl.5 FILE=G:\TPCD\DBS\HUNDGIG\TS_PARTS5.DBF
direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\parts.ctl log=k:\tpcd\log\hundgig\parts6.log
data=o:\tpcd\hundgig\data\part.tbl.6 FILE=H:\TPCD\DBS\HUNDGIG\TS_PARTS6.DBF
direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\parts.ctl log=k:\tpcd\log\hundgig\parts7.log
data=o:\tpcd\hundgig\data\part.tbl.7 FILE=I:\TPCD\DBS\HUNDGIG\TS_PARTS7.DBF
direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\parts.ctl log=k:\tpcd\log\hundgig\parts8.log
data=o:\tpcd\hundgig\data\part.tbl.8 FILE=J:\TPCD\DBS\HUNDGIG\TS_PARTS8.DBF
direct=true parallel=true
}
*wait
*time=End parts, Begin customer
*wait
*load
{
tpcd/tpcd control=k:\tpcd\ctl\customer.ctl log=k:\tpcd\log\hundgig\customer1.log
data=o:\tpcd\hundgig\data\customer.tbl.1
FILE=C:\TPCD\DBS\HUNDGIG\TS_CUSTOMER1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\customer.ctl log=k:\tpcd\log\hundgig\customer2.log
data=o:\tpcd\hundgig\data\customer.tbl.2
FILE=D:\TPCD\DBS\HUNDGIG\TS_CUSTOMER2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\customer.ctl log=k:\tpcd\log\hundgig\customer3.log
data=o:\tpcd\hundgig\data\customer.tbl.3
FILE=E:\TPCD\DBS\HUNDGIG\TS_CUSTOMER3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\customer.ctl log=k:\tpcd\log\hundgig\customer4.log
data=o:\tpcd\hundgig\data\customer.tbl.4
FILE=F:\TPCD\DBS\HUNDGIG\TS_CUSTOMER4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\customer.ctl log=k:\tpcd\log\hundgig\customer5.log
data=o:\tpcd\hundgig\data\customer.tbl.5
FILE=G:\TPCD\DBS\HUNDGIG\TS_CUSTOMER5.DBF direct=true parallel=true
}
*load
{
```

```
tpcd/tpcd control=k:\tpcd\ctl\customer.ctl log=k:\tpcd\log\hundgig\customer6.log
data=o:\tpcd\hundgig\data\customer.tbl.6
FILE=H:\TPCD\DBS\HUNDGIG\TS_CUSTOMER6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\customer.ctl log=k:\tpcd\log\hundgig\customer7.log
data=o:\tpcd\hundgig\data\customer.tbl.7
FILE=I:\TPCD\DBS\HUNDGIG\TS_CUSTOMER7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\customer.ctl log=k:\tpcd\log\hundgig\customer8.log
data=o:\tpcd\hundgig\data\customer.tbl.8
FILE=J:\TPCD\DBS\HUNDGIG\TS_CUSTOMER8.DBF direct=true parallel=true
}
*wait
*time=End customer, Begin supplier
*wait
*load
{
tpcd/tpcd control=k:\tpcd\ctl\supplier.ctl log=k:\tpcd\log\hundgig\supplier1.log
data=o:\tpcd\hundgig\data\supplier.tbl.1
FILE=C:\TPCD\DBS\HUNDGIG\TS_SUPPLIER1.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\supplier.ctl log=k:\tpcd\log\hundgig\supplier2.log
data=o:\tpcd\hundgig\data\supplier.tbl.2
FILE=D:\TPCD\DBS\HUNDGIG\TS_SUPPLIER2.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\supplier.ctl log=k:\tpcd\log\hundgig\supplier3.log
data=o:\tpcd\hundgig\data\supplier.tbl.3
FILE=E:\TPCD\DBS\HUNDGIG\TS_SUPPLIER3.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\supplier.ctl log=k:\tpcd\log\hundgig\supplier4.log
data=o:\tpcd\hundgig\data\supplier.tbl.4
FILE=F:\TPCD\DBS\HUNDGIG\TS_SUPPLIER4.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\supplier.ctl log=k:\tpcd\log\hundgig\supplier5.log
data=o:\tpcd\hundgig\data\supplier.tbl.5
FILE=G:\TPCD\DBS\HUNDGIG\TS_SUPPLIER5.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\supplier.ctl log=k:\tpcd\log\hundgig\supplier6.log
data=o:\tpcd\hundgig\data\supplier.tbl.6
FILE=H:\TPCD\DBS\HUNDGIG\TS_SUPPLIER6.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\supplier.ctl log=k:\tpcd\log\hundgig\supplier7.log
data=o:\tpcd\hundgig\data\supplier.tbl.7
FILE=I:\TPCD\DBS\HUNDGIG\TS_SUPPLIER7.DBF direct=true parallel=true
}
*load
{
tpcd/tpcd control=k:\tpcd\ctl\supplier.ctl log=k:\tpcd\log\hundgig\supplier8.log
data=o:\tpcd\hundgig\data\supplier.tbl.8
FILE=J:\TPCD\DBS\HUNDGIG\TS_SUPPLIER8.DBF direct=true parallel=true
}
*wait
*time=End supplier, Begin nation
*wait
*load
{
tpcd/tpcd control=k:\tpcd\ctl\nation.ctl log=k:\tpcd\log\hundgig\nation.log
data=o:\tpcd\hundgig\data\nation.tbl FILE=D:\TPCD\DBS\HUNDGIG\NATION.DBF
}
*wait
*time=End nation, Begin region
*wait
*load
{
tpcd/tpcd control=k:\tpcd\ctl\region.ctl log=k:\tpcd\log\hundgig\region.log
data=o:\tpcd\hundgig\data\region.tbl FILE=E:\TPCD\DBS\HUNDGIG\REGION.DBF
}
*wait
*time=End TPCD database load, End region
*wait
*wait
*bgoff
%e-dapop
%b-ixcre
*bgon=20
#########################################################################
###
# Index Creation Phase
*time=Begin TPCD analyze
*wait
*sql
{
shutdown
}
*wait
*sql
{
startup pfile=c:\tpcd\dbs\hundgig\p_analyze.ora
}
*wait
*sql
{
connect tpcd/tpcd;
analyze table lineitem estimate statistics sample 200000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze table orders estimate statistics sample 200000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze table partsupp estimate statistics sample 200000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze table parts estimate statistics sample 200000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze table customer estimate statistics sample 200000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze table supplier estimate statistics sample 2000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze table nation compute statistics;
}
*sql
{
connect tpcd/tpcd;
analyze table region compute statistics;
}
*wait
*time=End table analyze
*wait
*sql
{
shutdown
}
*wait
*sql
{
startup pfile=c:\tpcd\dbs\hundgig\p_run.ora
}
*wait
*time=Begin TPCD index creation
*wait
*sql
{
connect tpcd/tpcd;
```

```
drop index l_ored;
create index l_ored
on lineitem (l_orderkey,l_returnflag,l_extendedprice,l_discount)
pctfree 2
initrans 20
tablespace ts_l_ored
storage (initial 20m next 20m freelists 8 freelist groups 9 maxextents unlimited pctincrease
0)
unrecoverable
parallel (degree 16)
;
}
*wait
*time=End index creation: l_ored
*wait
*sql
{
connect tpcd/tpcd;
drop index l_pqesod;
create index l_pqesod
on lineitem (l_partkey,l_quantity,l_extendedprice,l_suppkey,l_orderkey,l_discount)
pctfree 2
initrans 20
tablespace ts_l_pqesod
storage (initial 20m next 20m freelists 8 freelist groups 9 maxextents unlimited pctincrease
0)
unrecoverable
parallel (degree 16)
;
}
*wait
*time=End index creation: l_pqesod
*wait
*sql
{
connect tpcd/tpcd;
drop index o_op;
create unique index o_op
on orders (o_orderkey,o_orderpriority)
pctfree 2
initrans 20
tablespace ts_o_op
storage (initial 25m next 25m freelists 8 freelist groups 9 maxextents unlimited pctincrease
0)
unrecoverable
parallel (degree 16)
;
}
*wait
*time=End index creation: o_op
*wait
*sql
{
connect tpcd/tpcd;
drop index ps_pksk;
create unique index ps_pksk
on partsupp (ps_partkey,ps_suppkey)
pctfree 2
tablespace ts_ps_pksk
storage (initial 20m next 20m maxextents unlimited pctincrease 0)
unrecoverable
parallel (degree 16)
;
}
*wait
*time=End index creation: ps_pksk
*wait
*sql
{
connect tpcd/tpcd;
drop index o_clokod;
create index o_clokod
on orders (o_clerk,o_orderkey,o_orderdate)
pctfree 2
initrans 20
tablespace ts_o_clokod
storage (initial 25m next 25m freelists 8 freelist groups 9 maxextents unlimited pctincrease
0)
unrecoverable
parallel (degree 16)
;
}
*wait
*time=End index creation: o_clokod
*wait
*sql
{
connect tpcd/tpcd;
drop index ps_spsa;
create unique index ps_spsa
on partsupp (ps_suppkey, ps_partkey, ps_supplycost,ps_availqty)
pctfree 2
tablespace ts_ps_spsa
storage (initial 20m next 20m maxextents unlimited pctincrease 0)
unrecoverable
parallel (degree 16)
;
}
*wait
*time=End index creation: ps_spsa
*wait
*sql
{
connect tpcd/tpcd;
drop index s_skey;
create unique index s_skey
on supplier (s_suppkey)
pctfree 2
tablespace ts_s_skey
storage (initial 20m next 20m maxextents unlimited pctincrease 0)
unrecoverable
parallel (degree 16)
;
}
*wait
*time=End index creation: s_skey
*wait
*sql
{
connect tpcd/tpcd;
drop index r_rn;
create unique index r_rn
on region (r_name,r_regionkey)
tablespace ts_r_rn
unrecoverable
parallel (degree 1)
;
}
*wait
*time=End index creation: r_rn
*wait
*sql
{
connect tpcd/tpcd;
alter table partsupp add primary key (ps_partkey,ps_suppkey) disable;
alter table partsupp enable primary key;
}
*wait
*time=End TPCD index creation
*wait
*bgoff
%e-ixcre
%b-anlyz
*bgon=20
##########################################################################
###
# Analyze Phase
*wait
*time=Begin TPCD analyze
*wait
*sql
{
shutdown
}
*wait
*sql
{
startup pfile=c:\tpcd\dbs\hundgig\p_analyze.ora
}
*wait
*time=Begin index analyze
*wait
*sql
{
connect tpcd/tpcd;
analyze index l_ored estimate statistics sample 200000 rows;
```

```
}
*sql
{
connect tpcd/tpcd;
analyze index o_op estimate statistics sample 200000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze index l_pqesod estimate statistics sample 200000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze index o_clokod estimate statistics sample 200000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze index ps_pksk estimate statistics sample 200000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze index s_skey estimate statistics sample 2000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze index ps_spsa estimate statistics sample 200000 rows;
}
*sql
{
connect tpcd/tpcd;
analyze index r_rn estimate statistics sample 200000 rows;
}
*wait
*time=End index analyze
*wait
*time=Begin alter storage parameters
*wait
*sql
{
connect tpcd/tpcd;
alter table lineitem deallocate unused;
alter table orders deallocate unused;
alter table lineitem storage (next 5m pctincrease 0);
alter table orders storage (next 1m pctincrease 0);
connect internal;
alter tablespace ts_supplier coalesce;
alter system switch logfile;
alter system switch logfile;
}
*wait
*sql
{
shutdown
}
*wait
*sql
{
host net stop oracleservicetpcd
host net stop oracletnslistener
host net start oracleservicetpcd
host net start oracletnslistener
startup pfile=c:\tpcd\dbs\hundgig\p_runf.ora
}
*wait
*time=End TPCD database preparation
*wait
*bgoff
%e-anlyz
```

CUSTOMER.CTL

```
---
---  customer.ctl for delimited records
---
options (direct = true, parallel = true)

unrecoverable

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

LINEITEM.CTL

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

options (direct = true, parallel = true)

unrecoverable

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

NATION.CTL

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

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

ORDERS.CTL

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

options (direct = true, parallel = true)

unrecoverable

load
into table orders
append
fields terminated by '|'
(
  o_orderkey            integer external,
  o_custkey             integer external,
  o_orderstatus         ,
  o_totalprice          float external,
```

```
  o_orderdate           date "yyyy-mm-dd",
  o_orderpriority       ,
  o_clerk               ,
  o_shippriority        integer external,
  o_comment
)
```

PARTS.CTL

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

options (direct = true, parallel = true)

unrecoverable

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

Partsupp.ctl

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

options (direct = true, parallel = true)

unrecoverable

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

REGION.CTL

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

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

SUPPLIER.CTL

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

options (direct = true, parallel = true)

unrecoverable

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

Appendix C: Query Text and Query Output

```
-- using default substitutions
-- Query 1 Original

SELECT
L_RETURNFLAG,
L_LINESTATUS,
SUM(L_QUANTITY) AS SUM_QTY,
SUM(L_EXTENDEDPRICE) AS SUM_BASE_PRICE,
SUM(L_EXTENDEDPRICE * (1 - L_DISCOUNT)) AS SUM_DISC_PRICE,
SUM(L_EXTENDEDPRICE * (1 - L_DISCOUNT) * (1 + L_TAX)) AS
SUM_CHARGE,
AVG(L_QUANTITY) AS AVG_QTY,
AVG(L_EXTENDEDPRICE) AS AVG_PRICE,
AVG(L_DISCOUNT) AS AVG_DISC,
COUNT(*) AS COUNT_ORDER
FROM LINEITEM
WHERE L_SHIPDATE <= TO_DATE('1998-12-01','YYYY-MM-DD') - 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            3773034.00       5319329289.68
5053976845.78
5256336547.68    25.51            35964.01         0.05
147907.00
N            F            100245.00        141459686.10
134380852.77
139710306.87     25.63            36160.45         0.05
3912.00
N            O            7464940.00       10518546073.98
9992072944.46
10392414192.06   25.54            35990.13         0.05
292262.00
R            F            3779140.00       5328886172.99
5062370635.93
5265431221.82    25.55            36025.46         0.05
147920.00


-- Query 2 (Original)

SELECT
S_ACCTBAL,
S_NAME,
N_NAME,
P_PARTKEY,
P_MFGR,
S_ADDRESS,
S_PHONE,
S_COMMENT
FROM PARTS, SUPPLIER, PARTSUPP, NATION, REGION
WHERE P_PARTKEY = PS_PARTKEY
AND S_SUPPKEY = PS_SUPPKEY
AND P_SIZE = 15
AND P_TYPE LIKE '%BRASS'
AND S_NATIONKEY = N_NATIONKEY
AND N_REGIONKEY = R_REGIONKEY
AND R_NAME = 'EUROPE'
AND PS_SUPPLYCOST =
(SELECT
MIN(PS_SUPPLYCOST)
FROM PARTSUPP PS1, SUPPLIER S1, NATION N1, REGION R1
WHERE PARTS.P_PARTKEY = PS1.PS_PARTKEY
AND S1.S_SUPPKEY = PS1.PS_SUPPKEY
AND S1.S_NATIONKEY = N1.N_NATIONKEY
AND N1.N_REGIONKEY = R1.R_REGIONKEY
AND R1.R_NAME = 'EUROPE')
ORDER BY S_ACCTBAL DESC, N_NAME, S_NAME, P_PARTKEY

S_ACCTBAL        S_NAME                    N_NAME
P_PARTKEY        P_MFGR
S_ADDRESS                                  S_PHONE
S_COMMENT
9828.21          Supplier#000000647        UNITED KINGDOM
13120.00         Manufacturer#5
jB16PyPyB7Bl52jMjSPw3mS                    33-258-202-4782
z1QhSiMj11Bm7COlLwh6Ql0B1R2Mg4CLn
LhiP0wiMzy72hlkP7l5in2y6RS6N130lz51nSRL5gOg5S26hPCCQN2L
9508.37          Supplier#000000070        FRANCE
3563.00          Manufacturer#1
M5C616R5h5SlMR3zzmLkSw24j2                 16-821-608-1166
m7z0CPShmBkhlChBAi3LkQ2CLw
mhl6QP362RPS3044CB2y41yhOhjlBin0CL7yhxmhS4hBM07kQ1yyjOjz3C
9508.37          Supplier#000000070        FRANCE
17268.00         Manufacturer#4
M5C616R5h5SlMR3zzmLkSw24j2                 16-821-608-1166
m7z0CPShmBkhlChBAi3LkQ2CLw
mhl6QP362RPS3044CB2y41yhOhjlBin0CL7yhxmhS4hBM07kQ1yyjOjz3C
9453.01          Supplier#000000802        ROMANIA
10021.00         Manufacturer#5
5yARQNSLNRAlOlBnkNQCik3SOlyClk7nmRhA2h0     29-342-882-6463
65y3RQ2i0OP6Nz7mS hC
PxwLy7L1jQy6Ol63xO3iBCz52Rm1zm0MziCMLij2n6wky51mBOwx Qh52iz
QB1545Amxyj
9453.01          Supplier#000000802        ROMANIA
13275.00         Manufacturer#4
5yARQNSLNRAlOlBnkNQCik3SOlyClk7nmRhA2h0     29-342-882-6463
65y3RQ2i0OP6Nz7mS hC
PxwLy7L1jQy6Ol63xO3iBCz52Rm1zm0MziCMLij2n6wky51mBOwx Qh52iz
QB1545Amxyj
9192.10          Supplier#000000115        UNITED KINGDOM
13325.00         Manufacturer#1
h0m3lzlSPMw2B0ny7LNyNCkjRRn7iyMlLBLA       33-597-248-1220
1QzQjhSyx
ixm2lgz2Ry7075RL3MS5z36x56hxmR0wLN0LBxm164LzCMmALzOAjn4kz7i4w
jOlCONl1C51M7nCMx66SBRAQA
9032.15          Supplier#000000959        GERMANY
4958.00          Manufacturer#4
205LNCzxMCnQ5gnz4n S3ynP6Mhnw              17-108-642-3106
Px z7kOx56l7jQz NwBBQhky yM7kLgxRQw5zw6 426Bm55lC6
OkQ7hQPLixjM7y47BNP16CRi0kjk354lgxh
8702.02          Supplier#000000333        RUSSIA
11810.00         Manufacturer#3
5iwkgN5n2BN15OmQk2602h0N6NzxPyiPN5lnj      32-508-202-6136
SgimAjmn3wL7Rlxmh3LCwOPnhjyl 7xxzxAN 4ACx43y65NwQ7P
8615.50          Supplier#000000812        FRANCE
10551.00         Manufacturer#2
h4i2M2O0 ky1g2mlBOmxjzj0hA2h6nkSNhP        16-585-724-6633
57i0NAyR0RP2jOh54C6B22OlSL
8615.50          Supplier#000000812        FRANCE

-- Query 3

SELECT
L_ORDERKEY,
SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT)) AS REVENUE,
TO_CHAR(O_ORDERDATE, 'YYYY-MM-DD'),
O_SHIPPRIORITY
FROM CUSTOMER, ORDERS, LINEITEM
WHERE C_MKTSEGMENT = 'BUILDING'
AND C_CUSTKEY = O_CUSTKEY
AND L_ORDERKEY = O_ORDERKEY
AND O_ORDERDATE < TO_DATE('1995-03-15','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

L_ORDERKEY       REVENUE
TO_CHAR(O_ORDERD
O_SHIPPRIORITY
260930.00        320547.25
1995-03-12
0.00
402497.00        298879.53
1995-02-12
0.00
457859.00        296490.68
1995-01-17
0.00
509889.00        294068.87
1995-02-03
0.00
58117.00         292632.83
1995-02-21
0.00
538311.00        279666.00
1995-03-07
0.00
588421.00        275477.12
1995-03-03
0.00
416167.00        273765.45
1995-02-22
0.00
97830.00         273227.06
1995-03-04
0.00
90276.00         272233.92
1995-03-04
0.00

-- Query 4 (Variant B)

SELECT
O_ORDERPRIORITY,
COUNT(*) AS ORDER_COUNT
FROM ORDERS
WHERE O_ORDERKEY IN
(SELECT
DISTINCT O_ORDERKEY
FROM LINEITEM, ORDERS
```

```
WHERE L_ORDERKEY = O_ORDERKEY
AND O_ORDERDATE >= TO_DATE('1993-07-01','YYYY-MM-DD')
AND O_ORDERDATE < ADD_MONTHS(TO_DATE('1993-07-01','YYYY-MM-
DD'),3)
AND L_COMMITDATE < L_RECEIPTDATE)
GROUP BY O_ORDERPRIORITY
ORDER BY O_ORDERPRIORITY

O_ORDERPRIORITY ORDER_COUNT
1-URGENT        999.00
2-HIGH          1002.00
3-MEDIUM        1021.00
4-NOT SPECIFIED 997.00
5-LOW           1089.00


-- Query 5

SELECT
N_NAME,
SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT)) AS REVENUE
FROM CUSTOMER, ORDERS, LINEITEM, SUPPLIER, NATION, REGION
WHERE C_CUSTKEY = O_CUSTKEY
AND O_ORDERKEY = L_ORDERKEY
AND L_SUPPKEY = S_SUPPKEY
AND C_NATIONKEY = S_NATIONKEY
AND S_NATIONKEY = N_NATIONKEY
AND N_REGIONKEY = R_REGIONKEY
AND R_NAME = '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
CHINA                     7349391.47
INDONESIA                 6485853.40
INDIA                     5505346.82
JAPAN                     5388883.59
VIETNAM                   4728846.60


-- Query 6

SELECT
SUM(L_EXTENDEDPRICE * L_DISCOUNT) AS REVENUE
FROM LINEITEM
WHERE L_SHIPDATE >= TO_DATE('1994-01-01','YYYY-MM-DD')
AND L_SHIPDATE < ADD_MONTHS(TO_DATE('1994-01-01','YYYY-MM-
DD'),12)
AND L_DISCOUNT BETWEEN .06 - 0.01 AND .06 + 0.01
AND L_QUANTITY < 24

REVENUE
11450588.04


-- Query 7 (Original)

SELECT
SUPP_NATION,
CUST_NATION,
YEAR,
SUM(VOLUME) AS REVENUE
FROM
(SELECT
N1.N_NAME AS SUPP_NATION,
N2.N_NAME AS CUST_NATION,
TO_CHAR(L_SHIPDATE,'YYYY') AS YEAR,
L_EXTENDEDPRICE * (1-L_DISCOUNT) AS VOLUME
FROM SUPPLIER, LINEITEM, ORDERS, CUSTOMER, NATION N1, NATION
N2
WHERE S_SUPPKEY = L_SUPPKEY
AND O_ORDERKEY = L_ORDERKEY
AND C_CUSTKEY = O_CUSTKEY
AND S_NATIONKEY = N1.N_NATIONKEY
AND C_NATIONKEY = N2.N_NATIONKEY
AND ((N1.N_NAME = '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, YEAR
ORDER BY SUPP_NATION, CUST_NATION, YEAR

SUPP_NATION               CUST_NATION
YEAR
REVENUE
FRANCE                    GERMANY
1995
4611421.44
FRANCE                    GERMANY
1996
4828420.37
GERMANY                   FRANCE
1995
6755766.84
GERMANY                   FRANCE
1996
5810951.40



-- Query 8 (Variant B)

SELECT
YEAR,
SUM(DECODE(NATION, 'BRAZIL', VOLUME, 0)) / SUM(VOLUME) AS
MKT_SHARE
FROM
(SELECT
TO_CHAR(O_ORDERDATE,'YYYY') AS YEAR,
L_EXTENDEDPRICE*(1-L_DISCOUNT) AS VOLUME,
N2.N_NAME AS NATION
FROM PARTS, SUPPLIER, LINEITEM, ORDERS, CUSTOMER, NATION N1,
NATION N2,
REGION
WHERE P_PARTKEY = L_PARTKEY
AND S_SUPPKEY = L_SUPPKEY
AND L_ORDERKEY = O_ORDERKEY
AND O_CUSTKEY = C_CUSTKEY
AND C_NATIONKEY = N1.N_NATIONKEY
AND N1.N_REGIONKEY = R_REGIONKEY
AND R_NAME = '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 YEAR
ORDER BY YEAR

YEAR
MKT_SHARE
1995
0.05
1996
0.09


-- Query 9 (Original)

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

NATION
YEAR
SUM_PROFIT
ALGERIA
1998
1946316.01
ALGERIA
1997
2973825.69
ALGERIA
1996
3308881.52
ALGERIA
1995
3092227.30
ALGERIA
1994
3406958.71
ALGERIA
1993
3140744.03
ALGERIA
1992
3330704.41
ARGENTINA
1998
3045410.01
ARGENTINA
1997
4255378.59
ARGENTINA
1996
4651751.94
```

```
-- Query 10

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

C_CUSTKEY        C_NAME                    REVENUE
C_ACCTBAL
N_NAME                    C_ADDRESS
C_PHONE
C_COMMENT
9722.00          Customer#000009722        464618.26
474.04
CANADA                    l Mwzn4NAk6j
13-518-602-8070
5L 50Oy RSgBAzPxmOSi5wk6xxOR7kh2nnPlgy7LBng2hOw5B01
RmCM120L24Pkg7PS1zwC11BCnz4L6i15PkixP26l66
12800.00         Customer#000012800        444265.64
1900.84
PERU                      57zjB3CQx4P4OB2R2MBi2mwhSllM4mn 4
nC6    27-142-205-3552
0hwglS77RB56Rx436lQ0N16CxhOPnmyhgwz
5z64wnj1kiC4jL350mM41y71hNxBllPjyA4hiN1wzjjM7SCxAN244mk2A
1025.00          Customer#000001025        442028.02
3363.46
INDIA                     lkiSn154M5ROi
18-588-456-4616
0Bl45z233Rniw0OO064nPBgP16kimO0y74iLh73g1N4 m310 jQ yQzPA50iC
3MA75g2Bj162Nw4P
13028.00         Customer#000013028        441692.24        -
452.66
UNITED KINGDOM            yP714ORSNgNN2LA3L5B
33-253-660-2127
xPkmnhL2BkhkNyww4khlxwwAymN h11PSjBCNMi50LkyOhO6CC
5nzOQCALzliOk2R66w l05hRPO3iSP
3694.00          Customer#000003694        438180.07
2960.44
UNITED KINGDOM            2CCklmCBOCC
33-421-331-3127
MzLxQxLlLx3MPxlAwg1B5kg61zxkPnk xiAm6PhMMAAQ2nzN3S6zzgP
x70w0lhhPx4QRzlMMy0204lAl3mBO7jh2jAP0N60wg367z
976.00           Customer#000000976        435897.63
7772.85
ROMANIA                    QzR 56Px1kgS  wANnAz02RS 30n Pm
29-436-660-4732
kzn32776 gwzkMzzzO4yxOAnkR7hR4R4x2SMwilz3x6h nN7OnNLRMml3
kz5SLwi1yklOxiwS4g0wmA5A 4hmgBSwRRiQ1
8206.00          Customer#000008206        429905.11
6046.36
ARGENTINA                 P yMg30BBBBx NMgC03AmzN2
11-571-859-1370
hLi122RMPmLC36Oy0kxO71zz2wCR0QQCl7z26hlQ3mM
13532.00         Customer#000013532        427731.80        -
924.18
KENYA
6ij7M5PBMx2kwwyz62Oj4SL5S0mRCwl3m1Rmw    24-525-332-7244
7ih7yRz214zO67AiNPx64nO5l5k yj6i3jLA5PCL15Q4QlA3ll60iM1P
iBxCixg6 1hCh2RCnjOzk5R OnO 1OhhC3m4631m5
12745.00         Customer#000012745        422327.69
9691.33
CHINA                     SgS1LMC4gB2NM3wh
28-985-189-6174
jl72wjSw0 S6 7L4Cgxw PkyO5Nl2LL7LBR
2344.00          Customer#000002344        411240.11
5597.22
MOROCCO                   O3PC7ikBgw OAzPAlm2P 426zm3BnBN6Q1O
6N   25-593-745-7663
5NBn0wRNngLw2z5kyn1AhL0ASyg6SMhM
i2kMOyxARAnlO0Q5j4CBNARix7ABlMAC


-- Query 11 (Original)

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) * 0.001000
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
12098.00         16227681.21
5134.00          15709338.52
13334.00         15023662.41
17052.00         14351644.20
3452.00          14070870.14
12552.00         13332469.18
1084.00          13170428.29
5797.00          13038622.72
12633.00         12892561.61
403.00           12856217.34


-- Query 12 (Variant B)

SELECT
L_SHIPMODE,
SUM(DECODE(O_ORDERPRIORITY,'1-URGENT',1,'2-HIGH',1,0)) AS
HIGH_LINE_COUNT,
SUM(DECODE(O_ORDERPRIORITY,'1-URGENT',0,'2-HIGH',0,1)) AS
LOW_LINE_COUNT
FROM ORDERS, LINEITEM
WHERE O_ORDERKEY = L_ORDERKEY
AND L_SHIPMODE IN ('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       654.00           950.00
SHIP       684.00           1004.00


-- Query 13 (Original)

SELECT
YEAR,
SUM(REVENUE) AS REVENUE
FROM
(SELECT
TO_CHAR(O_ORDERDATE,'YYYY') AS YEAR,
L_EXTENDEDPRICE * (1-L_DISCOUNT) AS REVENUE
FROM LINEITEM, ORDERS
WHERE  O_ORDERKEY = L_ORDERKEY
AND O_CLERK = 'Clerk#000000088'
AND L_RETURNFLAG = 'R'
) PERFORMANCE
GROUP BY YEAR
ORDER BY YEAR

YEAR
REVENUE
1992
1262855.73
1993
964121.03
1994
1750395.29
1995
198820.30


-- Query 14 (Variant C)

CREATE TABLE ALL_SALES0 (TYPE VARCHAR2(25), AMOUNT NUMBER(20,
2))
tablespace TS_SUPPLIER
pctfree 0
pctused 99
parallel 60
storage (initial 5m next 5m pctincrease 0)

Statement Processed in 0.09 seconds.

CREATE TABLE SUM_PROMO_SALES0 (PROMO_AMOUNT NUMBER(20, 2))
tablespace TS_SUPPLIER
pctfree 0
pctused 99
parallel 60
storage (initial 5m next 5m pctincrease 0)

Statement Processed in 0.04 seconds.

CREATE TABLE SUM_ALL_SALES0 (ALL_AMOUNT NUMBER(20, 2))
tablespace TS_SUPPLIER
pctfree 0
pctused 99
parallel 60
storage (initial 5m next 5m pctincrease 0)
```

```
Statement Processed in 0.04 seconds.

INSERT INTO ALL_SALES0
SELECT
P_TYPE, SUM(L_EXTENDEDPRICE *(1-L_DISCOUNT))
FROM  LINEITEM, PARTS
WHERE L_PARTKEY = P_PARTKEY
AND L_SHIPDATE >= TO_DATE('1995-09-01','YYYY-MM-DD')
AND L_SHIPDATE < ADD_MONTHS(TO_DATE('1995-09-01','YYYY-MM-
DD'), 1)
GROUP BY P_TYPE
Statement Processed in 3.20 seconds.

INSERT INTO SUM_PROMO_SALES0
SELECT
SUM(AMOUNT)
FROM  ALL_SALES0
WHERE TYPE LIKE 'PROMO%'
Statement Processed in 0.04 seconds.

INSERT INTO SUM_ALL_SALES0
SELECT
SUM(AMOUNT)
FROM ALL_SALES0
Statement Processed in 0.01 seconds.

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

PROMO_REVENUE
16.73

DROP TABLE ALL_SALES0

DROP TABLE SUM_PROMO_SALES0

DROP TABLE SUM_ALL_SALES0


-- Query 15 (Variant B)

CREATE TABLE REVENUE0 (SUPPLIER_NO INTEGER, TOTAL_REVENUE
NUMBER(20, 2))
tablespace TS_SUPPLIER
pctfree 0
pctused 99
parallel 60
storage (initial 5m next 5m pctincrease 0)

Statement Processed in 0.04 seconds.

INSERT INTO REVENUE0
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
Statement Processed in 2.96 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
389.00           Supplier#000000389
PB1Lx0xx6LMz3h7Rx63m6j3QmMx              34-885-883-5717
1418538.21

DROP TABLE REVENUE0


-- Query 16

SELECT
P_BRAND,
P_TYPE,
P_SIZE,
COUNT(DISTINCT PS_SUPPKEY) AS SUPPLIER_CNT
FROM PARTSUPP,PARTS
WHERE P_PARTKEY = PS_PARTKEY
AND P_BRAND <> 'Brand#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 '%Better Business Bureau%Complaints%')
GROUP BY P_BRAND, P_TYPE, P_SIZE
ORDER BY SUPPLIER_CNT DESC, P_BRAND, P_TYPE, P_SIZE

P_BRAND    P_TYPE                      P_SIZE
SUPPLIER_CNT
Brand#14   SMALL ANODIZED NICKEL      45.00            12.00
Brand#22   SMALL BURNISHED BRASS      19.00            12.00
Brand#25   PROMO POLISHED COPPER      14.00            12.00
Brand#35   LARGE ANODIZED STEEL       45.00            12.00
Brand#35   PROMO BRUSHED COPPER       9.00            12.00
Brand#51   ECONOMY ANODIZED STEEL     9.00            12.00
Brand#53   LARGE BRUSHED NICKEL       45.00            12.00
Brand#11   ECONOMY POLISHED COPPER    14.00            8.00
Brand#11   LARGE PLATED STEEL         23.00            8.00
Brand#11   PROMO POLISHED STEEL       23.00            8.00


-- Query 17 (Original)

SELECT
SUM(L_EXTENDEDPRICE)/7.0 AS AVG_YEARLY
FROM LINEITEM, PARTS
WHERE P_PARTKEY = L_PARTKEY
AND P_BRAND = 'Brand#23'
AND P_CONTAINER = 'MED BOX'
AND L_QUANTITY <
(SELECT
0.2* AVG(L1.L_QUANTITY)
FROM LINEITEM L1
WHERE L1.L_PARTKEY = P_PARTKEY)

AVG_YEARLY
24436.88
```

Appendix D: Seed and Query Substitution Parameters

-- using 697726 as a seed to the RNG
-- Query 1 Original

DELTA=91

-- Query 2 (Original)

SIZE=26
TYPE= BRASS
REGION= ASIA

-- Query 3

SEGMENT= FURNITURE
DATE=1995-03-08

-- Query 4 (Variant B)

DATE=1995-07-01

-- Query 5

REGION= ASIA
DATE=1994-01-01

-- Query 6

SHIPDATE=1995-01-01
DISCOUNT=0.04
QUANTITY=25

-- Query 7 (Original)

NATION1= JAPAN
NATION2= FRANCE

-- Query 8 (Variant B)

NATION= JAPAN
REGION= ASIA
TYPE= SMALL PLATED BRASS

-- Query 9 (Original)

COLOR= magenta

-- Query 10

DATE=1994-02-01

-- Query 11 (Original)

NATION=JAPAN
FRACTION=0.000001

-- Query 12 (Variant B)

SHIPMODE1=TRUCK
SHIPMODE2=AIR
DATE=1995-01-01

-- Query 13 (Original)

CLERK= Clerk#000000518

-- Query 14 (Variant C)

DATE=1995-09-01

-- Query 15 (Variant B)

DATE=1995-07-01

-- Query 16

BRAND= Brand#23
TYPE= LARGE BURNISHED
SIZE IN (27, 19, 16, 7, 44, 29, 30, 12)

-- Query 17 (Original)

BRAND= Brand#23
CONTAINER= MED CAN

Appendix E: Implementation Specific Layer/Source Code

qexecpl.c

```
/*=================================================================+
|      Copyright (c) 1996  Oracle Corp, Redwood Shores, CA        |
|                OPEN SYSTEMS PERFORMANCE GROUP                   |
|                   All Rights Reserved                           |
 +=================================================================+
| FILENAME
|    qexecpl.c
| DESCRIPTION
|    SQL Execution Engine, OCI version
| MODIFIED
|    pswong    04/02/96 - more polishing
|    pswong    03/25/96 - polish up
|    pswong        03/06/96 - created
 +=================================================================*/

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

#include "qexecpl.h"

/* Function Prototypes */

extern double gettime();

/* function prototypes from gen.c */

int get_statement();

/* Declare error handling functions */

void sql_error();
void sql_error2();

/* Other prototypes */

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

void usage() {

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



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

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

  /* Rollback just in case */

  orol(lda);

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

  SQLexit();

  exit(1);
}

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

  char msg[2048];

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

  fflush(stderr);
}
```

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

 int retcode;     /* Return code for get_statement */

 /* Initialize some variables */

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

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

 strcpy(logname, argv[1]);

 /* Process  optional parameters */

 argc -= 1;
 argv += 1;

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

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

SQLinit();

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

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

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

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

  switch (retcode) {

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

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

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

    /* Executes the query */
    SQLexec();

    s_tr_end = gettime();
    stmt_cnt++;

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

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

 /* Get Timing for the last query */

 tr_end = gettime();

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

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

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

 fflush(rep);
 fflush(logfile);

 /* Close the query template file */

 fclose(qtemp);

 /*  Disconnect from ORACLE. */

 SQLexit();
 exit(0);
}



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

void SQLinit() {

  int i;

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

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

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

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

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

  /* Open a cursor for us to use */

  OOPEN(&tpclda, &curq);

}
```

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

void SQLexec()
{
  int i;

  if (!end_flag) {

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

    tr_end = gettime();

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

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

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

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

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

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

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

      fflush(logfile);
      fflush(rep);

    }

    tr_start = tr_end;
    qry_cnt++;

    end_flag = 1;
  }

  s_tr_start = gettime();

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

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

  /* Prints the query text to the logfile */

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

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

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

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

  num_sel_list = define_output_variables();

  /* Executes the query and fetches the rows */

  (void) process_select_list(num_sel_list);


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

  rows_ret = curq.rpc;

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

  i=MAX_PREALLOC;

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

  /* reset set_fetchrows */

  num_to_fetch = -1;

}


void SQLexit() {

  int i;

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

  /* free all memory */

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

  /* Flush all output */

  fflush(rep);
  fflush(logfile);

}


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

int define_output_variables()
{

  int i;

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

    slist[i].buflen = MAX_COLNAME_SIZE;

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

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

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

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

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

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

    switch (slist[i].dbtype) {

    case NUM_TYPE:

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

      slist[i].dbsize = NUMWIDTH;

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

#ifdef HAVE_SCALE
      if (slist[i].scale != 0) {
                  defbuf = (double *) dlist[i]->fbuf;
```

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

    break;

  default:

    /* default is character string */

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

  }

  /* Define the column */

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

 }
 return i;
}


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

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

 int ntf;

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

 print_header(num);

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

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

 /* Fetch the rows and print them out */

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

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

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

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

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

}


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

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

 return 1;
}


int get_statement()
{

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

 /* Reset statement buffer */

 stmt[0] = '\0';

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

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

   /* remove blanks */

   str = line;

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

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

   strcat(stmt, str);

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

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

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


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

void *memalloc(size)
    int size;
{

 void *tmp;

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

```
}


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

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

  fprintf(logfile, "\n");

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

    /* format the output a little */

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

    /* do a little bit of formatting */

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

  fprintf(logfile,"\n");
}


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

  int i,j;
  int len;
  int cwid;

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

    len = 0;

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

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

      /* do a little bit of formatting */

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

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

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

void remove_newline(str)
    char *str;
{

  char *p;

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

qexecpl.h

```
/*===============================================================+
|      Copyright (c) 1995  Oracle Corp, Redwood Shores, CA      |
|                 OPEN SYSTEMS PERFORMANCE GROUP                |
|                    All Rights Reserved                        |
+===============================================================+
| FILENAME
|   qstream.h
| DESCRIPTION
|   SQL statement execution front-end header file.
| MODIFIED
|   pswong    03/07/96 - created
+===============================================================*/

#ifndef QSTREAMPL_H

#define QSTREAMPL_H

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

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

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

typedef struct cda_def csrdef;
typedef struct cda_def ldadef;

/* some basic definitions */

#define UNAME_LEN 64
#define MAX_FILE_PATH_LEN 128

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

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

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

/* defines and typedefs for parsing */

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

/* defines and typedefs for query description */

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

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

/* types for describe */

#define CHAR_TYPE 1
#define NUM_TYPE  2
```

```
#define INT_TYPE  3
#define FLT_TYPE  4
#define STR_TYPE  5
#define DATE_TYPE 12

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

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

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


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

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

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

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

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

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

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

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

extern int errno;

#define SQL_LEN 2048

#ifndef NULL
#define NULL 0
#endif

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

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#ifndef ub1
#define ub1 unsigned char
#endif

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

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

/* For get_statement */

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

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

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

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

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

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

#define ISOTXT "alter session set isolation_level = serializable"

#endif /* QSTREAMPL_H */
```

gtime.c

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

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

double gettime();

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

uf1.pc

```
/*=============================================================+
|      Copyright (c) 1995  Oracle Corp, Redwood Shores, CA     |
|                OPEN SYSTEMS PERFORMANCE GROUP                |
|                   All Rights Reserved                       |
+=============================================================+
| FILENAME
|    uf1.pc
| DESCRIPTION
|    TPC-D benchmark uf1 driver, PRO*C version
| MODIFIED
|    pswong   03/25/96 - more polish
|    pswong   02/28/96 - clean it up a little
|    pswong   04/21/95                - update to Spec 9.1, Use QGEN
+=============================================================*/

#include <stdio.h>
#include <string.h>
#include <setjmp.h>
/*#include <sys/param.h>*/
#include <errno.h>
#include <math.h>
#include <string.h>
```

```
/*#include <unistd.h>*/
#include <signal.h>
#include <stdlib.h>

#include "shared.h"

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

/* Include the SQL Communications Area. */

#include <sqlca.h>

/* Inclde the Oracle Communication Area */

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

extern double gettime();

/* Declare error handling functions */

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

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

/* batch size for inserts */

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

#ifndef NULL
#define NULL 0
#endif

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

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

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

/* declarations for ORDER */

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

/* declarations for LINEITEM */

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

EXEC SQL END DECLARE SECTION;

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

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

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

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

void usage() {

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

}



void sql_error()
{

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

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

}



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

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

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

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

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

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

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

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

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

  run_id  = atoi(argv[2]);

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

  proc_no = atoi(argv[3]);

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

  sf = atof(argv[4]);

  /* Process  optional parameters */
```

```
  argc -= 4;
  argv += 4;

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

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

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

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

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

  /* open those files */

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

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

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

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

  EXEC SQL WHENEVER SQLERROR DO sql_error();

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

  EXEC SQL CONNECT :lname;

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

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

  /* start doing the update */

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

  notdone = 1;   /* just in case */

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

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

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

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

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

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

    linecnt = get_item_line(ordcnt);

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

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

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

  /* end of this update function */

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

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

  EXEC SQL WHENEVER SQLERROR CONTINUE;

  /*  Disconnect from ORACLE. */

  EXEC SQL COMMIT WORK RELEASE;
```

```
  EXEC SQL WHENEVER SQLERROR DO sql_error();
  fclose(ordfile);
  fclose(itemfile);
  exit(0);
}


int get_ord_line () {

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

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

    /* extract columns from the line                    */
    /* Can't use fscanf here because the delimiter is | */

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

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

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

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

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

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

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

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

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


int get_item_line (rowcnt)
     int rowcnt;
{

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

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

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

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

    notyet = 1;

    while (notyet) {

      /* extract columns from the line */

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

      /* Compare the ORDERKEY values */

      if (okey == o_okey[j]) {

                l_okey[i] = okey;

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

      /* increments array index */
      i++;

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

```
  return (i);
}

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

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

  return (mark);
}


void wakeup() {
  return;
}
```

uf2_del.pc

```
/*=================================================================+
|      Copyright (c) 1995  Oracle Corp, Redwood Shores, CA        |
|                  OPEN SYSTEMS PERFORMANCE GROUP                 |
|                     All Rights Reserved                         |
 +=================================================================+
| FILENAME
|    uf1_del.pc
| DESCRIPTION
|    TPC-D benchmark uf2 driver, PRO*C version
|    Also used to restore uf1.
| MODIFIED
|    pswong    03/30/96 - polished version
|    pswong    06/13/95 - version 2
|    pswong        06/07/95 - created
 +=================================================================*/

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

#include "shared.h"

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

/* max length of SQL statement */

#define SQL_LEN 4096

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

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

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

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

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

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

#define             COMMIT_FREQ 4

/* Include the SQL Communications Area. */

#include <sqlca.h>

/* Inclde the Oracle Communication Area */

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

extern double gettime();

/* Declare error handling functions */

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

#ifndef NULL
#define NULL 0
#endif

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

EXEC SQL BEGIN DECLARE SECTION;

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

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

EXEC SQL END DECLARE SECTION;

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

FILE *ordfile;           /* flatfile for ORDERS                 */


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

void usage() {

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


void sql_error()
{

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

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

}



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

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

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

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

  /* signal(SIGUSR1, wakeup); */

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

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

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

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

  run_id  = atoi(argv[2]);

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

  proc_no = atoi(argv[3]);

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

  sf = atof(argv[4]);

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

  u_flag = atoi(argv[5]);

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

  /* Process  optional parameters */

  argc -= 5;
  argv += 5;

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

  /* open the files */

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

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

  EXEC SQL WHENEVER SQLERROR DO sql_error();

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

  EXEC SQL CONNECT :lname;

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

  /* pause(); */

  /* start the restore from uf1 */

  /* delete from ORDERS, hold the commit */

  notdone = 1;   /* just in case */

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

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

    /* Execute the deletes, hold commit */

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

    /* Commit once in a while */

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

    orows += o;
    lrows += l;

  }

  /* COMMIT all deletes */

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

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

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

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

  /* end of the restore */

  EXEC SQL WHENEVER SQLERROR CONTINUE;

  /*  Disconnect from ORACLE. */

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


int get_ord_line (void) {

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

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

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

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

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

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


void wakeup() {
  return;
}
```

RUNUF1.SH

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

O=$ORACLE_HOME
TPCD_DIR=/tpcd

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

export UPDATE_DIR

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


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

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

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

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

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

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

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

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

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

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

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

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

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

i=1
PID=""

# perform the update function 1

START=`$GTIME`

while [ $i -le $PAR ]
do

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

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

done

# now wait for start signal (SIGKILL will do)

# ./trig

# If we haven't timed out, start everyone

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

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

wait

END=`$GTIME`

# Done

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

RUNUF2.SH

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

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

export UPDATE_DIR

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

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

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

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

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

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

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

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

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

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

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

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

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

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

i=1
```

```
PID=""

START=`$GTIME`

# perform the update function

while [ $i -le $PAR ]
do

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

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

done

# now wait for start signal (SIGKILL will do)

#./trig

# If we haven't timed out, start everyone

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

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

wait

END=`$GTIME`

# Done

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

runpower1.sh

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

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

RUN_ID_FILE=${TPCD_HOME}/r_id

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

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

OUT=${TPCD_HOME}/out

GTIME=${SRC_DIR}/gtime

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

# The defaults

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

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


usage () {

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

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

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

SF=$1
PARA=$2
THROUGHPUT=1

if [ $SF -eq 1 ]
  then
QRY_DIR=${TPCD_HOME}/queries/q_one_0.sql
INIT_DIR=${TPCD_HOME}/dbs/onegig
elif [ $SF -eq 3 ]
  then
QRY_DIR=${TPCD_HOME}/queries/q_three_0.sql
INIT_DIR=${TPCD_HOME}/dbs/three
elif [ $SF -eq 10 ]
  then
QRY_DIR=${TPCD_HOME}/queries/q_ten_0.sql
INIT_DIR=${TPCD_HOME}/dbs/tengig
elif [ $SF -eq 100 ]
  then
```

```
QRY_DIR=${TPCD_HOME}/queries/q_hundred_0.sql
INIT_DIR=${TPCD_HOME}/dbs/hundgig
elif [ $SF -eq 300 ]
  then
QRY_DIR=${TPCD_HOME}/queries/q_threehund_0.sql
INIT_DIR=${TPCD_HOME}/dbs/threehund
elif [ $SF -eq 1000 ]
  then
QRY_DIR=${TPCD_HOME}/queries/q_tbyte_0.sql
INIT_DIR=${TPCD_HOME}/dbs/tbyte
fi

export SF PARA THROUGHPUT

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

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

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

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

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

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

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

# Execute UF1

echo "Start UF1 at `date`"

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

# Execute Query Stream

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

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

# Execute UF2

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

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

echo "END UF2 `date`"

END=`$GTIME`

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

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

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

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


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

echo resetting database `date`
sh ${UPD_SPT}/resuf1.sh -p ${U2PROG} -u ${USER} ${RUN_ID} ${SF} 1 ${PARA} \
                > ${OUT}/resuf1.${RUN_ID}.${HID} 2>&1
sh ${UPD_SPT}/resuf2.sh -p ${U1PROG} -u ${USER} ${RUN_ID} ${SF} 1 ${PARA} \
                > ${OUT}/resuf2.${RUN_ID}.${HID} 2>&1
echo done resetting database `date`
```

Appendix F: Initial Ten Rows

```
L_SHIPDAT L_ORDERKEY L_DISCOUNT L_EXTENDED L_SUPPKEY  L_QUANTITY L L_PARTKEY  L L_TAX      L_COMMITD L_RECEIPT L_SHIPMODE L_LINENUMB
L_SHIPINSTRUCT           L_COMMENT
--------- ---------- ---------- ---------- ---------- ---------- - ---------- - ---------- --------- --------- ---------- ---------- ---------- ----------------------------------------
01-APR-93     184001 .029999999   29261.25     934202         25 A    14684187 F .059999999 31-MAY-93 28-APR-93 TRUCK               1 NONE                    RjlMgkM4j37wOnCA50 N
gRi4kl10yMO
08-AUG-96     183941        .01 15719.4004     369504         10 N    11369503 O .029999999 28-SEP-96 22-AUG-96 MAIL              4 COLLECT COD             1m11Sl1
liMw1P5mii7AR6APA0N0mhwj
22-APR-97     183942 .100000001 61295.9609     235459         44 N     7235458 O        .07 14-APR-97 16-MAY-97 AIR               1 COLLECT COD
AMjhwikjMmxBMSxkj6kO01gLLRkPQ6BL1Lj
30-MAY-97     183942 .029999999 19852.9805     831400         14 N     6331387 O .079999998 24-MAY-97 12-JUN-97 SHIP              2 DELIVER IN PERSON
2zO7wwMzmCkAhMy3wiLBBQygx34
26-MAR-97     183942 .100000001 47794.8789     362380         38 N    11612346 O .050000001 18-MAY-97 06-APR-97 TRUCK             3 NONE                    lhSNP53C06k
2CSSzB066ih6QLL
17-MAR-97     183942 .029999999 38835.1602     936410         28 N     8686401 O .039999999 02-JUN-97 24-MAR-97 REG AIR           4 NONE                    2L5l4wnP5ng1O076k0C
20-MAY-97     183942        .01   26235.25     325026         25 N    12325025 O .029999999 27-APR-97 12-JUN-97 SHIP              5 COLLECT COD             l43SR2hgCy7
18-JUN-97     183942 .039999999 50167.3008     688631         31 N     6688630 O        .02 09-APR-97 25-JUN-97 MAIL              6 COLLECT COD             77wn R6yx4hyCL
06-OCT-96     183943          0 53063.1211     792470         34 N    15792469 O .059999999 13-OCT-96 10-OCT-96 REG AIR           1 DELIVER IN PERSON       6P QC5Cm
S037BgSwhMwB33j1mhRwB0 L
28-NOV-96     183943 .059999999 40235.6016      14093         40 N     4014092 O        .01 11-NOV-96 11-DEC-96 FOB               2 TAKE BACK RETURN        h4Lm2mS4m55CQLl
01745wh7 BPjBN7
10 rows selected.

C_CUSTKEY  C_MKTSEGME C_NATIONKE C_NAME                  C_ADDRESS                      C_PHONE         C_ACCTBAL  C_COMMENT
---------- ---------- ---------- ------------------------ ------------------------------ --------------- ---------- ----------------------------------------------------------------
   1875001 BUILDING           17 Customer#001875001       BziPQP0yAzyl2S1                27-230-790-2550    8681.43
mjL54N07gOC0mx3ANgg2Sg32QRBj0zjzMQA32112ziOmhwSl
   1875002 MACHINERY          10 Customer#001875002       yM3PiMCz3AByPgRj7PBwB3Q        20-658-206-2716    2827.31 0CABA60ymA
ihk0zx4mxlCPN3hN5ALSBSnNA6mMC5C PPO32CCL
   1875003 HOUSEHOLD           5 Customer#001875003       6y1BRO7M111z3L4MjQwQllL7MQCwiM7z6350jx  15-295-782-5405    7543.69 7zg3y4jMBNRgAkz27kmCCw 0zyO
wL73MxN5MNAyRgmCC1zwS2ljyPOMjS0S6CCPLyM2626m6g7mi5j5
   1875004 FURNITURE          20 Customer#001875004       zmh7g3kxzOzjlMx10              30-565-417-7846    9910.4  BgORkNP4PB3nSS31Pl471yk2inPRznLgSmL
   1875005 AUTOMOBILE         22 Customer#001875005       Cm5llx14B                      32-739-551-8731    163.44 w1 h773Sz5OwABMy5jB4SxPO64QlkO
0CRL7kNyyRy3hOOA00igjAjAiwm1N0
   1875006 AUTOMOBILE          5 Customer#001875006       jghgyPwNLPxgORhCw2L65h5Mi55i AwRN      15-933-468-8568    5788.04 SMgijC3i lRkhNn5CSjSB5y1Lmiw3L4Pz
lyCOLww3lh2i7mRyj1Nxy0wMml6Nk2g
   1875007 BUILDING           18 Customer#001875007       R3iSnxPw0 6lO3ONgwkg5x5hh0Qi5P CxxL5PPLk 28-829-506-9685    4624.35
3NjRi4nhN4wCmAiLSl3w5ywkNnL74lw4mMASR
   1875008 BUILDING           10 Customer#001875008       0wkx5lAR32QOQjNwnQn            20-202-964-2155    4455.7 hL7QA 23P14Bngxwi
hCNmyMRg62yzBmxAz7lSCwNwSCAPAngx
   1875009 HOUSEHOLD          12 Customer#001875009       A44 61giS16 kCOQR4yjkS 27mLCjmA0       22-191-857-3836    7974.42 0L2z4lB1kCSjmwL ynOOLSNyzSlxRR0hB6l
LnzSL h mnmkkQyh1NLLiBNASiOigMkihiCB7z
   1875010 BUILDING            0 Customer#001875010       yC04mOQyRz6ny                  10-888-280-6176    85.36 k AAkQgLhL4jwCkjw0P2mA0MnRB3gSgijMN
10 rows selected.

O_ORDERDA O_ORDERKEY O_CUSTKEY  O_ORDERPRIORITY O_SHIPPRIO O_CLERK        O O_TOTALPRI O_COMMENT
--------- ---------- --------- --------------- ---------- --------------- - ---------- ---------------------------------------------------------------------------------------
01-OCT-96     540426   11214214 1-URGENT                 0 Clerk#000036545 O 81687.4063 xLLPQ751hO24N0jjiQin4A1QzMQR2Cw5 lQBQ2501gPw55wz76z47lnB
Cw1wSA5znRBP1Aky
08-FEB-97     540427    2288485 5-LOW                    0 Clerk#000002905 O 163704.484  NBOhQP3g16 0i1SOCi6PRMQyOnm5xCCj4x5h1P6l
21-DEC-92     540428    4605661 4-NOT SPECIFIED          0 Clerk#000008461 F 116785.938 4OLP24yCNyh1NwO233ARkS
07-JUL-95     540429    1989509 3-MEDIUM                 0 Clerk#000000969 O 31850.4297 Rj RQnA37zlPMmiS2NxMA0iOS2xzklnMON01mii4xR14SN13SzLSSLAzQ6lmQMkR
21-MAY-96     540430    5488396 3-MEDIUM                 0 Clerk#000071920 O 32356.7695 5nR1Sy7j37BBOR021L2BROR7k wnSwnmnO4lAP6BON40n70
29-AUG-97     540431    3682289 4-NOT SPECIFIED          0 Clerk#000048934 O 285370.281 6lQnBlOzy20njliSOO65A 65wM
04-OCT-96     540456   13217324 2-HIGH                   0 Clerk#000032774 O 112350.031 i j BmPMCQPkj xxlyki5Lzilzmg0mLn2ClhxQ0SP yjlm
06-MAR-96     540457    8550169 4-NOT SPECIFIED          0 Clerk#000031439 O   340607.5  7QNOARkjmAl51AkN3h yzCMw
05-MAR-92     540458    2680901 1-URGENT                 0 Clerk#000094983 F 246540.656 hgR16jz677A7My7AjihOkPzjx6MQnLM3gx
01-JAN-93     540459   12897694 1-URGENT                 0 Clerk#000062995 F 224197.797 yA3RgSyOCzLgAhygNCgCjAPP
10 rows selected.

PS_PARTKEY PS_SUPPKEY PS_SUPPLYC PS_AVAILQT PS_COMMENT
---------- ---------- ---------- ---------- ----------------------------------------------------------------------------------------
   7500001     500002     558.27       2788 zgnx7i54PSSMOxnk4Ck zkNLlwhBBOQQ2zRNnhl6C6h5PklQ3 S2AyyNBkPP53APACw2iw4BR062RL6g
   7500001     750009     422.43        376 N0iRl wPkyPQnSym 1 1C23MjRxjBy1AMzBgn22N33ijylkll6SARxM5R5PR42g4y2wA6ShwnnOz3glQ
   7500001         16      28.47       6823 6LBM1QxPCRkn2xx7L5x3jPBOPLS4LhgSSCww4  kz3Bl3AkxMPwBl6MPz6PSBgh7k0hjNx26Qhli1lCN
   7500001     250023     308.53       1912 MPLOCPNiLzCQO4LmBmy 3wxQMN3P1khi OQ0n6nB71SOk22R5
   7500002     500003     781.63       3233 O1wxAkiShAC506zm1Ammhwj1kwOCRLkMMCg7ky 3ROy0l5n66ClP3l0k 5zLl26h SkSljk 5P253
   7500002     750010     142.97       6735 jhxNiw0M4n6gMhhxx62CBnm5CB6BAin5S6lk3k4QB457QxQzAm
   7500002         17     588.25       5682 MzPk4hj546OzA4iP4gjCSQk76BMl3zCO7Ln2Bn2jPSLBwyNBhMmliNOBA1A7iyy 4mLCmLhRQ6iMkxPg

   7500002     250024     810.14       8896 wNmMmLhhLLzByRPBiLLji3CC1zPM5n7MO34jn40MnlnjRQzkQnBlRSkPlnm3zjL0Sk2l20R653 17BN
   7500003     500004     717.77       6513 Lmz1wy23xgBl167AR 1RO343BQinnBz 6jni15zhy0MO2Cwx3l7O4Rgi7xj1M4mlhjQ1BljzlM3hhQSm
   7500003     750011     743.01       3654 Nx6Aw h5Bk1Bh42OjzlRgChwBkwA7SQjC2xOLChymmiNB3Qj15wRN3xAmQ3M1y7ROP035RA0QCQ140OB
10 rows selected.

P_PARTKEY  P_TYPE                 P_SIZE     P_BRAND    P_NAME                                   P_CONTAINE P_MFGR                P_RETAILPR P_COMMENT
---------- ------------------------- ---------- ---------- ---------------------------------------- ---------- ------------------------- ---------- ----------------------
```

```
   3123452 MEDIUM PLATED COPPER            48 Brand#15   moccasin dark goldenrod chartreuse dim            SM DRUM    Manufacturer#1            1475.3
hS0Nijm3i134w6x1AO2jh
   3123453 SMALL BURNISHED COPPER          19 Brand#32   slate tan almond red deep                         LG JAR     Manufacturer#3            1476.3 xlxSAx4
   3123454 ECONOMY BRUSHED COPPER          22 Brand#11   sienna blue blush gainsboro peach                 SM JAR     Manufacturer#1            1477.3 S62xPyN16LRil0S
   3123455 PROMO PLATED NICKEL              1 Brand#15   forest moccasin burlywood rose maroon             LG CAN     Manufacturer#1            1478.3
giP4xSAM46Pn3O47kMCj
   3123456 PROMO POLISHED NICKEL           18 Brand#13   cream cornsilk black rosy mint                    WRAP JAR   Manufacturer#1            1479.3 kn256
   3123457 LARGE ANODIZED STEEL             5 Brand#44   chocolate lawn beige pale navajo                  LG PKG     Manufacturer#4            1480.3  0lCzg46Rhn AO4 nwCSS
   3123458 ECONOMY ANODIZED NICKEL         47 Brand#23   white firebrick violet dim cornflower             JUMBO CASE Manufacturer#2            1481.3 CnhzBzjM
   3123459 ECONOMY BRUSHED NICKEL           4 Brand#23   steel indian metalic white antique                LG BOX     Manufacturer#2            1482.3 L PygRkjklMx
   3123460 PROMO BRUSHED BRASS              8 Brand#51   lawn purple forest chocolate dark                 LG BOX     Manufacturer#5            1483.31 zi6iCS 33C0QLL0
   3123461 STANDARD ANODIZED BRASS         27 Brand#51   royal chocolate aquamarine forest almond          JUMBO DRUM Manufacturer#5            1484.31 4NiLj6zL4kN
AmOPPlA
10 rows selected.

S_SUPPKEY  S_NATIONKE S_COMMENT                                                         S_NAME               S_ADDRESS                      S_PHONE        S_ACCTBAL
---------- ---------- ---------------------------------------------------------------- ------------------------- ------------------------------ --------------- ----------
    686408          0 Ahi1nBMgPw4CCxP11Ml1lASCgj1lmm17Rmyw4SxO33nQQ7 55BRxi6zL         Supplier#000686408        lw 66iPzkkmmO1j                10-950-921-
9859     8598.2
    686409          8 0O6iQiCC7jhhMw3h3w3ji1L5Mx AjO5P45RP56OB zl4P2k4BSygl7jLRQ4L1kl0l7 5RSyj       Supplier#000686409        04winN5inkxwlLB                18-115-
629-7836     996.47
    686410         22 CniMxjlAOzLzjRkSBAzRRmQwOzy x1SRQgAQzBgy3znknP6iPAS0jS42CN6z4QNyim5xOCy7nm26A4NR Supplier#000686410        7wgOjQLh3Cg
32-204-220-7167    3610.15
    686411         15 On7h45lhkLjPyOjC3QhB02RLP6SRxkBMjOkz6 6zwS5khjR3NO6jiMn7PxMwwyCP7g3POS7jyySl7x   Supplier#000686411        nl5iLCk2 NNB R
25-995-816-2332    8833.85
    686412         22 nn6zSPPQQnjMR1PA 33xwi50yQ503l335z                                Supplier#000686412        O3mSgnyLyB57 hyCQ              32-640-215-9937     1660.5
    686413          1 gnlyjxhy5R2lAzkxCx05N6MhB0gzzk7hkRL4m5ixRNMi11                    Supplier#000686413        j70BCm6il2MPN zjymnj1Qk        11-218-227-8279
-961.94
    686414         12 204R2x43l36zOh4gPmh5 i7w1QL4O                                     Supplier#000686414        35MxigB5Q2QywB                 22-278-129-9310     645.38
    686415         12  Rgjj74k7wi6jjB7BxiCS2SPx24PLh2j5ASQg                             Supplier#000686415        7kBjSSLyCk6nRNkhi0             22-321-748-5229     9760.65
    686416          7 MLgm5i02Ahil Sxi 7mPCgmzy2Rn3miRn302Ozg3kSS1Cyjx3R0               Supplier#000686416        L57LQmLxz6PzkAgz3P77P BS4nkn1Shyz2     17-721-
225-5849    2270.03
    686417         24 z313lw4Ry3CwzO1BN6N16m4l5i3jlL5hA3SynRkz2CkR5m4Qmxywizyz17nClhP6jQQlmMygLnO77QRO Supplier#000686417        S2M54LRgwk
34-645-580-3624    2295.44
10 rows selected.

N_NATIONKE N_NAME                 N_REGIONKE N_COMMENT
---------- ------------------------- ---------- -------------------------------------------------------------------------------
         0 ALGERIA                            0 2Cxhl7 L1iwk6hMh300izngN32CPwCikyLk6khMzSRA
         1 ARGENTINA                          1 zQn3Okwz1wLn7PLS3OhCgn56kP5PyRikgi1B7lL
         2 BRAZIL                             1 gLmS0nACAmnBCj2klki7RCPNgPxnCOjNg4k OiAg57COSOm1NwCnOyLx40R SC y20gPPAkNk5hxRhR5
         3 CANADA                             1  4yMO AhnQ5Lh wzQAM662Aw1ByCl7CxmzRwNR5nAlO4 x
         4 EGYPT                              4 l1im5126  Cxj NMQmLxOikni02j2m3Ah4yNR1QQiL507j2QSlyN
         5 ETHIOPIA                           0 NS7n LSOP Oz5n1AlB2S02nN0lMh4SBxP iRhBO 047R26 2BlM
         6 FRANCE                             3 3mjmizl S 3L3k2hNNhNlP4w370xRxyN15wn
         7 GERMANY                            3 z nOP4RkwO CmzBB 516mAg lByw4OM3QyNPA
         8 INDIA                              2 MNlR5RCiRMj1l11wjN7Myn M1lylNlMmBQl7PL4C  kkxQkgPQ7i3w6B67R2QkOO40xl4Q2iw76jRL7i
         9 INDONESIA                          2 SjPmQO71Lj 7ABj6MxlAQk3nLwi73BPxzCwjzMn4zlLzgg6nnz0j0w zxC66gP6ykRPMg
10 rows selected.

R_REGIONKE R_NAME                 R_COMMENT
---------- ------------------------- -------------------------------------------------------------------------------
         0 AFRICA                    xSx31zz31Cl1z4OAnmm05AjiOxC3AMMNOgC0kACgwngg3glP7LLLywlQy7R
         1 AMERICA                     kgyh3LSnC72k6zlAz0LP3k2L4QB1QL1O673OjO1SPj0ngQ7CO100SBgmgRQ4lgPCMk21A425iklyAR4y
         2 ASIA                      NSg6xlMlA1lzm6mOR0Ajx nhRA77NgRxBwL1M6Py RjySB3RLwkyPkwMM2R1BQ xAzkOgkjml0gAghi
         3 EUROPE                    zlSL7Qwg12hMBL5lhlz0M45QkjShwSyiO04MLOh7wn1ARLQPyPAyAiil576l1Li7AlnR1S RQ4SLny7B
         4 MIDDLE EAST                 RllxmhPLz3Cy2mNlg4QMBnNASM ACki MPki7Oi
5 rows selected.
```

Appendix G: Price Quotes

Compaq

APR-30-1997 14:57 P.03

Advanced Solutions

9777 W. Gulf Bank, #8 Suite 1000
Houston, Tx 77040-3113
Phone (713) 849-2828
Fax (713) 849-2850

Item Description	Stock Number	Price Each	QTY	Total	5 Year Service	Total Price
Proliant 6000 6/200 512K Model 1X	273350-001	$14,337.00	1	$14,337.00	$5,017.95	$19,354.95
Proliant 6000 Internal Drive Cage	273314-B21	$432.00	1	$432.00	$151.20	$583.20
Rack External Keyboard	187344-001	$32.00	1	$32.00	$11.20	$43.20
Keyboard/Mouse/Monitor Extension Ca	169989-001	$76.00	1	$76.00	$26.60	$102.60
42U 19" Rack	165753-001	$1,620.00	1	$1,620.00	$567.00	$2,187.00
22U 19" Rack	163747-001	$1,458.00	1	$1,458.00	$510.30	$1,968.30
Rack Sidewall Kit (For 42U Rack)	165652-001	$200.88	1	$200.88	$70.31	$271.19
Proliant 6000 Rack Mount Kit	273316-B21	$492.00	1	$492.00	$172.20	$664.20
6/200 - 512K Processor Expansion Kit	273315-B21	$2,905.00	3	$8,715.00	$3,050.25	$11,765.25
1 GB DIMM Kit	241774-B21	$26,460.00	2	$52,920.00	$18,522.00	$71,442.00
SMART-2/P SCSI Array Controller	194753-001	$2,138.40	5	$10,692.00	$3,742.20	$14,434.20
Compaq VGA V50 - 15" Color Monitor	264150-001	$383.00	1	$383.00	$134.05	$517.05
Compaq UPS M2500	163760-001	$1,446.00	2	$2,892.00	$1,012.20	$3,904.20
Rack Mount Proliant Storage System	272800-001	$1,620.00	8	$12,960.00	$4,536.00	$17,496.00
4.3 GB Pluggable SCSI-2 Drive	146742-006	$1,227.00	1	$1,227.00	$429.45	$1,656.45
9.1 GB Pluggable SCSI-2 Drive	199882-001	$2,376.00	58	$137,808.00	$48,232.80	$186,040.80
4\16GB TurboDAT Drive	142181-001	$1,084.00	1	$1,084.00	$379.40	$1,463.40
MKS Toolkit for NT	N/A	$499.00	1	$499.00	5 Yr WTY	$499.00
Microsoft Visual C++ Subscription Svc.	N/A	$499.00	1	$499.00	$2,495.00	$2,994.00
Microsoft Windows NT Server v. 4.0	N/A	$809.00	1	$809.00	$4,045.00	$4,854.00
Microsoft Resource Kit for NT 4.0	N/A	$499.00	1	$499.00	Inc Above	$499.00
			Totals:	$249,634.88	$93,105.11	$342,739.99

Prices include a large volume discount and may vary when items are purchased seperately.
Service prices based upon a service contract for all items listed.
The above quotation is valid for 60 days.

Jim Cockrill
Systems Consultant
(713) 849-2828

ORACLE

Oracle Corporation
New England Development Center
One Oracle Drive
Nashua
New Hampshire 03062

May 1, 1997

Mr. Mike Nikolaiev
Compaq Computer Corporation
PO Box 692000
Houston, TX 77269
Fax: 281-514-8375

Dear Mike:

In response to your inquiry, here is the U.S. pricing information you requested for the specified configuration:

Oracle7 License	$ 35,880
Oracle Silver support for 5 years	$ 48,960

Sincerely,

Jan Schwartz