



ORACLE®

TPC Benchmark™ D
Full Disclosure Report

Sun Starfire™ Ultra™ Enterprise™ 10000

Using

Oracle7

Submitted for Review

March 21, 1997

TPC Benchmark D Full Disclosure Report

Release 7.3.4

First Printing

© 1997 Sun Microsystems, Inc.
2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A.

All rights reserved. This product and related documentation are protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the United States Government is subject to the restrictions set forth in DFARS 252.227-7013 (c)(1)(ii) and FAR 52.227-19, Rights in Technical Data and Computer Software (October 1988).

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

TRADEMARKS

Sun, Sun Microsystems, the Sun logo, Ultra Enterprise 10000, Starfire, SMCC, the SMCC logo, SunSoft, the SunSoft logo, Solaris, SunOS, OpenWindows, DeskSet, ONC, and NFS are trademarks or registered trademarks of Sun Microsystems, Inc. All other product names mentioned herein are the trademarks of their respective owners.

All SPARC trademarks, including the SCD Compliant Logo, are trademarks or registered trademarks of SPARC International, Inc. SPARCstation, SPARCserver, SPARCengine, SPARCworks, and SPARCcompiler are licensed exclusively to Sun Microsystems, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK® and Sun™ Graphical User Interfaces were developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

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

ORACLE7, SQL*DBA, SQL*Loader, SQL*Net and SQL*Plus are registered trademarks of Oracle Corporation.

Veritas is a registered trademark of Veritas Corporation.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.

Sun Microsystems, Inc., believes that the information in this document is accurate as of its publication date. The information in this document is subject to change without notice. Sun Microsystems, Inc., assumes no responsibility for any errors that may appear in this document.

The pricing information in this document is believed to accurately reflect prices in effect on January 22, 1997. However, Sun Microsystems and Oracle Corporation provide no warranty on the pricing information in this document.

The performance information in this document is for guidance only. System performance is highly dependent on many factors including system hardware, system and user software, and user application characteristics. Customer applications must be carefully evaluated before estimating performance. Sun Microsystems, Inc., does not warrant or represent that a user can or will achieve a similar performance. No warranty on system performance or price/performance is expressed or implied in this document.

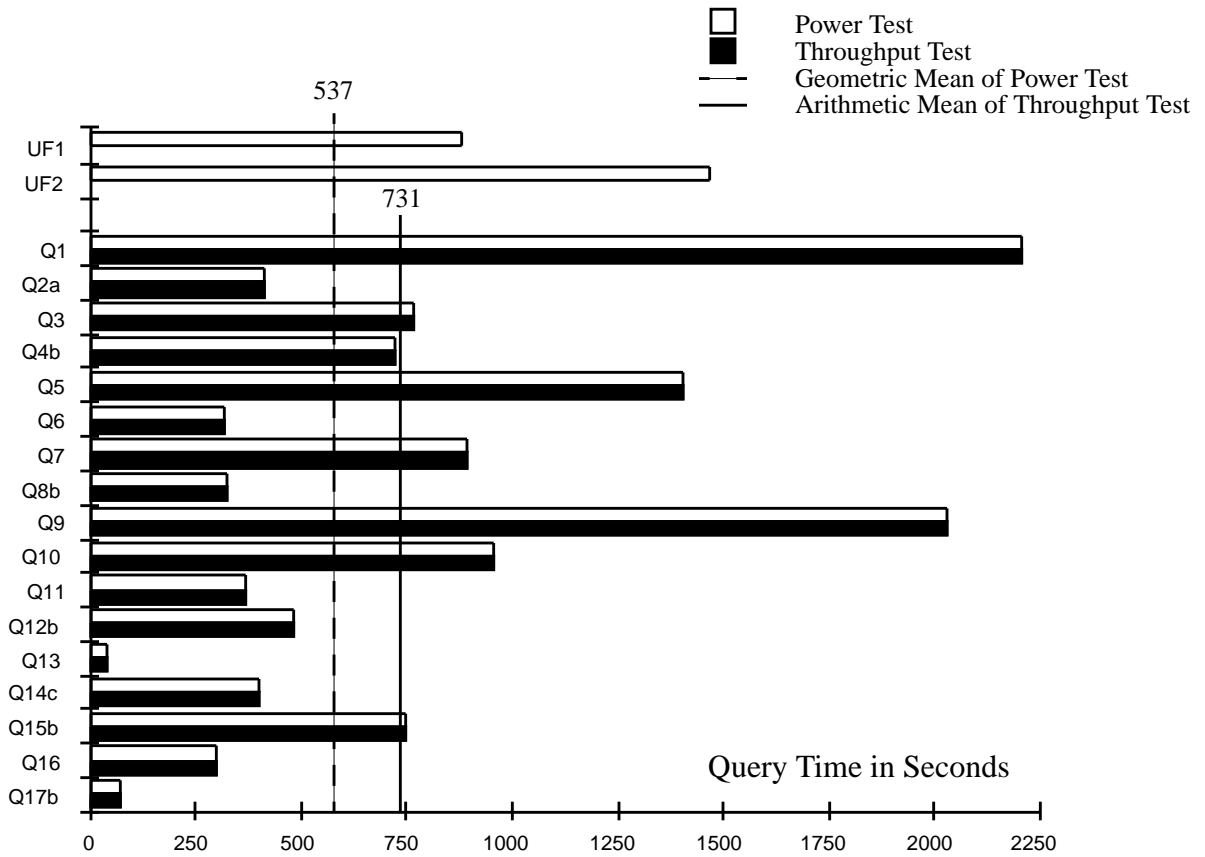


Sun Starfire Ultra Enterprise 10000 using Oracle7

TPC-D 1.2.1

Report Date:
March 21, 1997

Total System Cost	TPC-D Power	TPC-D Throughput	Price/Performance	
\$5,254,624	2009.5 QppD@300GB	1241.9 QthD@300GB	\$3,326 QphD@300GB	
Database Size	Database Manager	Operating System	Other Software	Availability Date
300GB	Oracle7 v7.3.3	Solaris 2.5.1	Veritas Volume Manager 2.3	May 1, 1997 See Note on next page



Database Load Time = 15:09:10 hours

Disk Size/Database Size = 6.04

RAID : Y

System Configuration:

- 64 250Mhz UltraSPARC 64-bit CPUs with 1MB Cache
- 16384 MB memory
- 61 SPARCStorage RSM with 7 x 4.2GB drives each
- 1 (2 x 9 Gbyte) Differential/Wide Disk Tray with 2 x 9 Gbyte drives



Sun Starfire Ultra Enterprise 10000 using Oracle7

TPC-D 1.2.1

Report Date:
March 21, 1997

Part Number	Description	Qty	Unit Price	Extended Price	Maintenance per Mo/Yr	5 Yr. Total Maintenance
Server Hardware: Pricing from CAT Technology, Inc.						
E10000	E10000 Base Cabinet	1	142500.00	142,500.00	7,181.00	344,688.00
2760A	CPU System Board, unpop	16	45000.00	720,000.00	772.00	592,896.00
2530A	UltraSPARC 250Mhz 1MB cache	64	12000.00	768,000.00		
2750A	SSP	1	6671.25	6,671.25	123.00	5,904.00
7025A	Memory Board, unpopulated	16	7500.00	120,000.00		
7022A	256MB for Ex000	64	3562.50	228,000.00		
783A	Elite 9 Tray, 2 x 9Gbyte Disks	1	8175.00	8,175.00		
6504A	RSM Tray (7*4.2GB Disk)	61	12150.00	741,150.00		
2730A	Dual SBus IO Board	16	5625.00	90,000.00		
1062A	SBus DiFF Fast/Wide SCSI-2	62	971.25	60,217.50		
9681A	Power Control Module	1	750.00	750.00		
9671A	Fan Tray	16	1500.00	24,000.00		
3800A	Power Cord for North America	4	0.00	0.00		
956A	I/O Expansion Cabinet	6	6000.00	36,000.00	224.00	64,512.00
979A	12 Meter SCSI Cable	62	198.75	12,322.50		
3800A	Power Cord for Expansion Cab	6	0.00	0.00		
3875A	AC Breaker & Filter	4	2250.00	9,000.00		
9685A	48 Volt Power Supply	8	3000.00	24,000.00		

ALL HARDWARE SUBTOTAL **2,990,786.25** **1,008,000.00**

Server and Client Software: Sun Software Pricing from Sun

CC-P	SPARC Compiler Single Usr. License	1	695.00	695.00	20.00	1,140.00
E10000-SW-AP	Software for AP	1	0.00	0.00		0.00
E10000-SW-SSP	Software for SSP	1	0.00	0.00		0.00
SOLS-C	Solaris 2.5.1 on CD-ROM	1	0.00	0.00		0.00

Sun Software Subtotal: □ **695.00** **20.00** **1,140.00**

Oracle pricing from Oracle Corp.

□	Oracle7 License	1	574,080.00	574,080.00		783,360.00
---	-----------------	---	------------	-------------------	--	-------------------

ALL SOFTWARE SUBTOTAL **574,775.00** **784,500.00**

Discount

Maintenance Volume Discount	5%	-50,457.00
Yearly Prepayment Discount	5%	-52,979.85

TOTAL HARDWARE & SOFTWARE COST (5YR) **5,254,624.40**

QphD **1,579.78**
\$/QphD **3326.17**

Sun hardware has 1-year warranty bundled.
During the warranty period, maintenance is discounted 50%
for parts that require an upgrade to their standard warranty

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.

Audited by: Francois Raab, Information Paradigm, Inc.

Note: Hardware available NOW
Oracle Server software available on May 1, 1997



**Sun Starfire
Ultra Enterprise 10000
using Oracle 7**

TPC-D 1.2.1

**Report Date:
March 21, 1997**

Numerical Quantity Summary

Measurement Results:

Database Scaling Factor (SF/Size)	= 300
Total Data Storage / Database Size	= 6.04
Database Load Time	= 15:09:10 hours
Query Streams for Throughput Test	= 0
TPC-D Power Metric (QppD@300GB)	= 2009.5
TPC-D Throughput Metric (QthD@300GB)	= 1241.9
Composite QphD@300GB	= 1579.8
Total System Price Over 5 Years	= \$5,254,624
TPC-D Price/Performance Metric	= \$3,326

Measurement Intervals:

Measurement Interval in Throughput Test (Ts)	= 14,784 Seconds
Duration of Stream Execution:	

Stream ID	Seed	Start Date	Start Time	End Date	End Time
Stream00	1110820981	Friday March 7	21:55:01 PST	Friday March 7	1:22:17 PST
UF1		Friday March 7	21:40:19 PST	Friday March 7	21:55:01 PST
UF2		Saturday March 8	1:22:17 PST	Saturday March 8	1:46:42 PST

TPC-D Timing Intervals (in seconds)

Q1	Q2a	Q3	Q4b	Q5	Q6	Q7	Q8b	Q9	Q10
2202.8	414.4	769.1	721.9	1404.2	316.0	890.6	324.1	2029.8	953.8

Q11	Q12b	Q13	Q14c	Q15b	Q16	Q17b	UF1	UF2
370.1	484.7	40.7	399.0	745.0	297.0	71.8	882.6	1465.6



Information Paradigm

TPC TRANSACTION PROCESSING
PERFORMANCE COUNCIL

Certified Auditor

Test Sponsors: Karl Haas
Manager, Server Performance
Oracle Corporation
100 Oracle Parkway
Redwood Shores, CA 94065

Brad Carlile
Manager, Strategic Apps. Eng.
Sun Microsystems, Inc.
8300 SW Creekside Place
Beaverton, OR 97008

March 17, 1997

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

Platform: Sun Starfire Ultra Enterprise 10000
DataBase Manager: Oracle7 Version 7.3.3
Operating System: Solaris Version 2.5.1

The results were:

CPU's Speed	Memory	Disks	QppD@300GB	QthD@300GB
Sun Ultra Enterprise 10000				
64 x UltraSPARC (250 MHz)	1MB L2 per cpu 16,384 MB	427 x 4.2 GB 2 x 9.0 GB	2009.5	1241.9

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

- The TIME table was not used
- The input variables were generated by QGEN
- The database was populated using DBGEN
- The database was maintained by the "Reset" method
- The throughput metric was computed using the results from the power test
- The ratio between the longest and the shortest query was such that no adjustment was necessary
- A compliant implementation specific layer was used
- The query text was produced using compliant variants and minor modifications

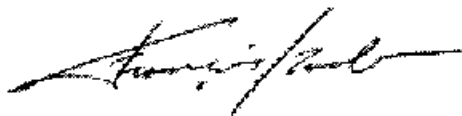
1373 North Franklin Street • Colorado Springs, CO 80903-2527 • Office: 719/473-7555 • Fax: 719/473-7554

-
-
- The database records were defined with the proper layout and size
 - The database was properly scaled to 300GB and populated accordingly
 - The database load time was correctly measured and reported
 - The ACID Properties were verified and met
 - The reported execution times were correctly measured and reported
 - Measurement repeatability was verified
 - At least 8 hours of database log was configured
 - The system pricing was verified for major components and maintenance
 - The major pages from the FDR were verified for accuracy

Additional Audit Notes:

None.

Respectfully Yours,



François Raab
President

Sun Ultra Enterprise 10000

1373 North Franklin Street • Colorado Springs, CO 80903-2527 • Office: 719/473-7555 • Fax: 719/473-7554

TPC Benchmark D Overview

The TPC Benchmark™ D (TPC-D) is a Decision Support benchmark. It is a suite of business-oriented queries and concurrent updates. The queries and the data populating the database have been chosen to have broad industry-wide relevance while maintaining a sufficient degree of ease of implementation. This benchmark illustrates Decision Support systems that:

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

TPC-D evaluates the performance of various Decision Support systems by the execution of sets of queries against a standard database under controlled conditions. The TPC-D queries:

- Give answers to real-world business questions
- Are far more complex than most OLTP transactions
- Include a rich breadth of operators and selectivity constraints
- Generate intensive activity on the part of the database server component of the system under test
- Are executed against a database complying to specific population and scaling requirements
- Are implemented with constraints derived from staying closely synchronized with an on-line production database

1. General Items

1.1 Benchmark Sponsor

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

Sun Microsystems and Oracle Corporation are the sponsors of this TPC-D benchmark.

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*

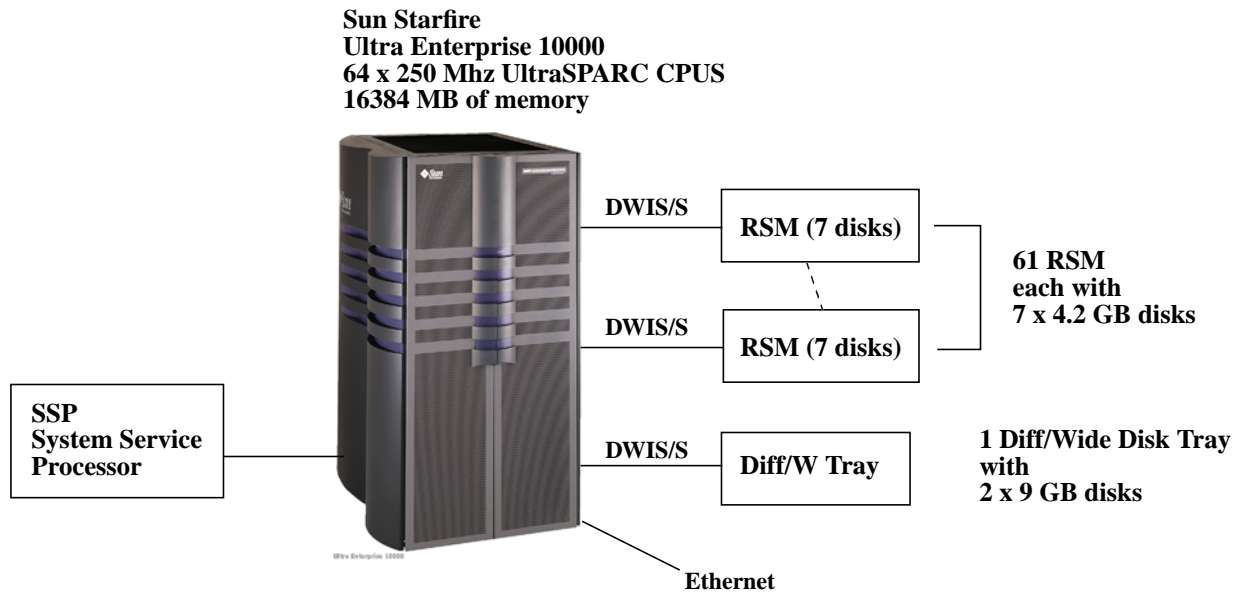
This requirement can be satisfied by providing a full list of all parameters and options.

Appendix A contains the Solaris and Oracle7 parameters used in this benchmark.

1.3 Configuration Diagram

Provide diagrams of both the measured and priced configurations, accompanied by a description of the differences. This includes, but is not limited to:

- *Number and type of processors*
- *Size of allocated memory, and any specific mapping/partitioning or memory unique to the test*
- *Number and type of disk drive 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, client processes, transaction monitors, software drivers, etc.)*



Tested and priced configurations are identical.

2. Clause 1 Logical Database Design

2.1 Database Definition Statements

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

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

2.2 Physical Organization

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

No record clustering or order clustering was used. Column ordering has been modified for some tables. Refer to the table create statements in Appendix B for further details.

2.3 Horizontal Partitioning

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

No horizontal partitioning was used.

2.4 Replication

While there are some restrictions placed upon physical replication of objects in the test and qualification databases (see Clause 1.5.6), any such replication must be disclosed.

No replication was used.

3. Clause 2 Queries and Update Functions

3.1 Query Language

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

SQL was the query language used to implement all queries.

3.2 Verifying Method for Random Number Generation

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

DBGEN version 1.1.0C and QGEN version 1.1.0D were used to verify random numbers for this TPC-D benchmark.

3.3 Generating Values for Substitution Parameters

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

The supplied QGEN version 1.1.0D was used to generate the substitution parameters.

3.4 Query Text and Output Data from Qualification Database

The executable query text used for query validation must be disclosed along with the corresponding output data generated during the execution of the query text against the qualification database. (If minor modifications (see Clause 2.2.3) have been applied to any functional query definitions or approved variants in order to obtain executable query test, 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.

3.5 Query Substitution Parameters and Seeds Used

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

Appendix D contains the seed and query substitution parameters.

3.6 Query Isolation Level

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

Level 1 isolation was used to run the queries.

3.7 Source Code of Update Functions

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 layer/driver code included in Appendix E.

3.8 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 uses the reset option. Source code for this is included in Appendix E.

4. Clause 3 Database System Properties

4.1 Atomicity

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

4.1.1 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 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.
3. The ACID Transaction 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. It was verified that the appropriate rows had 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 4.2.1.

1. The consistency of the ORDER and LINEITEM tables was verified based on a sample of O_ORDERKEY's.
2. 100 ACID Transactions were submitted from each of 2 execution streams.
3. The consistency of the ORDER and LINEITEM tables was reverified.

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. The 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. The 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. Another ACID Transaction, T2, was started using the same O_KEY and L_KEY and a randomly selected DELTA.
 3. T2 waited.
 4. T1 was allowed to COMMIT and T2 completed.
 5. It was verified that $T2.L_EXTENDEDPRICE = T1.L_EXTENDEDPRICE + (DELTA1*(T1.L_EXTENDEDPRICE/T1.L_QUANTITY))$

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

4.3.5 Concurrent Progress of Read and Write Transactions

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

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

4.3.6 Read-Only Query Conflict with Update Transaction

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

1. A Transaction, T1, was started which executed Q1 (from Clause 2.3) against the qualification database, with DELTA = 0.
2. An ACID Transaction, T2, was started for a randomly selected O_KEY, L_KEY and DELTA.
3. T2 completed.

-
4. T1 completed subsequently.
 5. It was verified that appropriate rows in ORDER, LINEITEM and HISTORY tables were changed.

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 TPC-D tables and log files were mirrored. During the durability test the disk containing one side of a data file mirror was removed from its cabinet. Similarly the disk containing one side of a log file mirror was removed from its cabinet. The test continued uninterrupted, using the remaining side of the mirror.

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 and memory failure tests were combined. Power to the server was turned off during the durability test. When power was restored, the system rebooted and the database was restarted. The durability success file and the HISTORY table were compared and the counts 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.

5. Clause 4 Scaling and Database Population

5.1 Ending 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	Rows
Order	450,000,000
Lineitem	1,799,989,091
Customer	45,000,000
Part	60,000,000
Supplier	3,000,000
Partsupp	240,000,000

Table	Rows
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.

Refer to Appendix F for details of the database layout.

5.3 Database partition/replication mapping

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

The database was neither partitioned nor replicated.

5.4 RAID feature

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

Table/Index	RAID type
tables	RAID 1 + 0
Indexes	RAID 1
temp tablespace	RAID 1
log	RAID 1 + 0
usrtmp tablespace	RAID 1

5.5 Modifications to the DBGEN

Any modifications to the DBGEN (see clause 4.2.1) source code must be disclosed. In the event that a program other than DBGEN was used to populate the database, it must be disclosed in its entirety.

The supplied DBGEN version 1.1.0C was used to generate the database population for this benchmark.

5.6 Database Content of Initial Ten Rows

The content of the first 10 rows of each table in the test database must be disclosed.

Appendix G contains the first 10 rows of each table in the test database.

5.7 Database Load Time

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

The database load time was 15 hours 9 minutes 10 seconds.

5.8 Data Storage Ratio

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

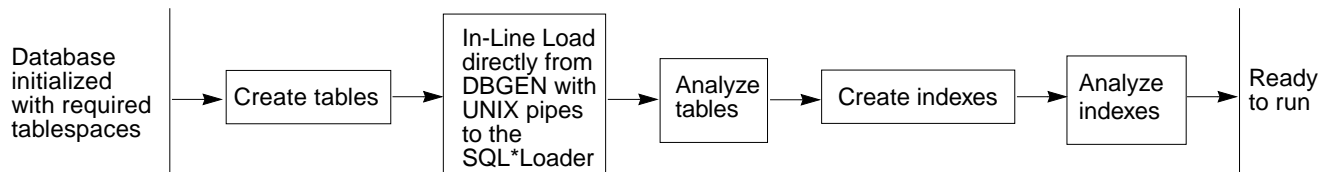
The data storage ratio is computed from the following information:

Disk Type	# of Disks	Space per Disk	Sub-Total Disk Space
RSM	329	4.2 GB	1793 GB
Diff/Wide	2	9.0 GB	18 GB
		Total Space	1811 GB
		Data Storage Ratio	6.04

5.9 Database Load Mechanism Details and Illustration

The details of the database load mechanism must be described and illustrated with a block diagram.

The database was loaded using in-line data generation all on the tested and priced configurations. The SQL*Loader read the data from UNIX pipes that were feed by DBGEN program. For Lineitem the output of DBGEN was split into 64 pipes, which were loaded into 128 datafiles. The rest of the tables were split into 64 pipes except for NATION and REGION. On the validation database, the data was generated into one pipe and then loaded into 2 datafiles for each of the tables, except for NATION and REGION..



6. Clause 5 Performance Metrics and Execution Rules

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. System boot
2. Database Restart
3. UF1 Update Transaction
4. Stream 00 Execution
5. UF2 Update Transaction

6.2 Timing Intervals for Each Query and Update Functions

The timing intervals 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 contained in the Numerical Quantity 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 and End Date/Times for 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 contained in the Numerical Quantity Summary in the Preface of this document.

6.5 Total Elapsed Time of the Measurement Interval

The total elapsed time of the measurement interval.

The total elapsed time of the throughput test was 14,784 seconds.

6.6 Update Function Start Date/Time and Finish Date/Time

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 Quantity Summary in the Preface of this document.

6.7 Timing Intervals for Each Query and Each Update Function for Each Stream

The timing intervals for each query and each update function for each stream 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 Quantity Summary in the Preface of this document.

6.8 Performance Metrics

Verify that the metrics are computed as required.

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

6.9 Reproducibility Method

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

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

Run ID	QppD@300GB	QthD@300GB	QphD@300GB
Run 1	2009.5	1241.9	1579.8
Run 2	2014.6	1252.3	1588.4
Difference	Less than 1%	Less than 1%	Less than 1%

7. Clause 6 SUT and Driver Implementation

7.1 Driver

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

The Power Test is performed by a shell script called runpower1.sh. QGEN is first called with a stream id of 0 to generate the QET for the Power Test. UF1 is then started by executing the runuf1.sh script. Query submission follows, with the qexec.ott ISL program. The execution of the UF2 script runuf2.sh rounds out the Power Test execution. Both wall-clock and high-resolution times are collected for all measurement intervals.

7.2 Implementation-Specific Layer

If any 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 which submits the query to the SUT.

The ISL program (qexec.ott) utilizes 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 (qexec.ott) 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.

7.3 Update Function

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

8. Clause 7 Pricing

8.1 Hardware and Software Used

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

Refer to 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 \$5,254,624. For details of pricing, see the second page of the Executive Summary.

Hardware prices are from reseller price quotes. Refer to Appendix H.

8.3 Availability Date

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

All hardware components and the operating system used in this benchmark are available now.

Oracle v. 7.3.3 will be available May 1, 1997.

9. Auditor's Information and Attestation Letter

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

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

Appendix A. Solaris and Oracle7 Parameters

Oracle7 parameters:

```
Sat Mar 8 09:34:55 PST 1997
control_files =
(/export/tpcd/sf300/dbs/controltpcd1.ct1,
/export/tpcd/sf300/dbs/controltpcd2.ct1)

background_dump_dest =
/export/tpcd/sf300/rdbms/log
core_dump_dest = /export/tpcd/sf300/dbs
user_dump_dest =
/export/tpcd/sf300/rdbms/log

db_name = sf300
db_block_size = 16384

db_files = 1022

always_anti_join = hash
async_read = TRUE
async_write = FALSE
audit_trail = FALSE
background_dump_dest =
/export/tpcd/sf300/rdbms/log
user_dump_dest =
/export/tpcd/sf300/rdbms/log
ccf_io_size = 131072
compatible = 7.3.3
cpu_count = 64
cursor_space_for_time = FALSE
db_block_buffers = 20000
db_block_lru_latches = 8
db_block_size = 16384
db_file_multiblock_read_count = 64
db_file_simultaneous_writes = 32
db_files = 1020
db_name = sf3000
db_writers = 300
delayed_logging_block_cleanouts = TRUE
distributed_transactions = 0
dml_locks = 2560
enqueue_resources = 64000
global_names = FALSE

hash_area_size = 47185920
hash_multiblock_io_count = 128
log_archive_buffer_size = 32
log_archive_start = FALSE
log_buffer = 1677721
log_checkpoint_interval = 999999999999
log_checkpoints_to_alert = TRUE
log_simultaneous_copies = 512
log_small_entry_max_size = 0
max_dump_file_size = 70000000
max_rollback_segments = 300
open_cursors = 500
optimizer_mode = CHOOSE

optimizer_percent_parallel = 100
optimizer_search_limit = 8
parallel_max_servers = 384
parallel_min_servers = 384
parallel_server_idle_time = 1
processes = 1020
recovery_parallelism = 32
rollback_segments=(t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14,t15,t16,t17,t18,t19,t20,t21,t22,t23,t24,t25,t26,t27,t28,t29,t30,t31,t32,t33,t34,t35,t36,t37,t38,t39,t40,t41,t42,t43,t44,t45,t46,t47,t48,t49,t50,u1,u2,u3,u4,u5,u6,u7,u8,u9,u10,u11,u12,u13,u14,u15,u16,u17,u18,u19,u20,u21,u22,u23,u24,u25,u26,u27,u28,u29,u30,u31,u32,u33,u34,u35,u36,u37,u38,u39,u40,u41,u42,u43,u44,u45,u46,u47,u48,u49,u50,v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16,v17,v18,v19,v20,v21,v22,v23,v24,v25,v26,v27,v28,v29,v30,v31,v32,v33,v34,v35,v36,v37,v38,v39,v40,v41,v42,v43,v44,v45,v46,v47,v48,v49,v50)
```

```
serializable = FALSE
sessions = 1000
shared_pool_size = 350000000
sort_area_size = 4194304
sort_direct_writes = TRUE
sort_read_fac = 15
sort_write_buffer_size = 65536
sort_write_buffers = 8
sql92_security = FALSE
transactions = 500
transactions_per_rollback_segment = 1
use_readv = FALSE
use_ism = TRUE

v733_plans_enabled = FALSE
Sat Mar 8 09:34:55 PST 1997
```

Solaris parameters:

```
*ident "@(#)system1.15
92/11/14 SMI" /* SVR4 1.5 */
*
* SYSTEM SPECIFICATION FILE
*
* moddir:
*
* Set the search path for modules. This has a
format similar to the
* csh path variable. If the module isn't found in
the first directory
* it tries the second and so on. The default is
/kernel /usr/kernel
*
* Example:
* moddir: /kernel /usr/kernel
/other/modules

* root device and root filesystem configuration:
*
* The following may be used to override the defaults
provided by
* the boot program:
*
* rootfs: Set the filesystem type of the root.
*
* rootdev: Set the root device. This should be a
fully
* expanded physical pathname. The default is the
physical pathname of the device where the boot
program resides. The physical pathname is
highly platform and configuration dependent.
*
* Example:
* rootfs:ufs
*
* rootdev:/sbus@1,f8000000/esp@0,800000/sd@3,0:a
*
* (Swap device configuration should be specified in
/etc/vfstab.)

* exclude:
*
* Modules appearing in the moddir path which are NOT
to be loaded,
* even if referenced. Note that `exclude' accepts
either a module name,
* or a filename which includes the directory.
*
* Examples:
* exclude: win
* exclude: sys/shmsys

* forceload:
*
* Cause these modules to be loaded at boot time,
(just before mounting
* the root filesystem) rather than at first
```

```

reference. Note that
*   forcload expects a filename which includes the
directory. Also
*   note that loading a module does not necessarily
imply that it will
*   be installed.
*
*   Example:
*       forcload: drv/foo

* set:
*
*   Set an integer variable in the kernel or a module
to a new value.
*   This facility should be used with caution. See
system(4).
*
*   Examples:
*
*   To set variables in 'unix':
*
*       set nautopush=32
*       set maxusers=40
*
*   To set a variable named 'debug' in the module
named 'test_module'
*
*       set test_module:debug = 0x13
* SUNWecsr start
* The following system parameters are inserted here in
order to
* override the default values used by workstations.
* These values have been tuned for better performance on
Enterprise
* server systems.
* Changed by rcl 12/9/96: the default for 16 GB sys was
32343 -> 1/4 GByte
set lotsfree=0x500
* DO NOT EDIT!
set tune_t_gpptslo=250
set tune_t_minarmem=100
set tune_t_minasmem=250
set msgsys:msginfo_msgmap=200
set msgsys:msginfo_msgmni=100
set msgsys:msginfo_msgttl=80
set msgsys:msginfo_msgseg=2048
set shmsys:shminfo_shmmax=2042626048
* set shmsys:shminfo_shmmax=268435456
set shmsys:shminfo_shmseg=200
set semsys:seminfo_semmmap=100
set semsys:seminfo_semmni=1000
set semsys:seminfo_semmns=4000
set semsys:seminfo_semmnu=800
set semsys:seminfo_semmns1=512
set semsys:seminfo_semume=100
set pln:pln_enable_detach_suspend=1
set soc:soc_enable_detach_suspend=1
set ssd:ssd_enable_detach_suspend=1
* SUNWecsr end

* vxvm_START
forcload: drv/vxio
forcload: drv/vxspec
* vxvm_END

set report_ce_log=1
set report_ce_console=1

set maxphys=1048576
set autoup=600

```

```

Memory size: 16384 Megabytes
CPU Units: Frequency Cache-Size Version
A: MHz MB Impl. Mask B: MHz MB Impl.
Mask
Mask
C: MHz MB Impl. Mask D: MHz MB Impl.
-----
Board 0: 250 1.0 11 1.1 250 1.0 11 1.1
1.1 250 1.0 11 1.1 250 1.0 11
Board 1: 250 1.0 11 1.1 250 1.0 11
1.1 250 1.0 11 1.1 250 1.0 11
Board 2: 250 1.0 11 1.1 250 1.0 11
1.1 250 1.0 11 1.1 250 1.0 11
Board 3: 250 1.0 11 1.1 250 1.0 11
1.1 250 1.0 11 1.1 250 1.0 11
Board 4: 250 1.0 11 1.1 250 1.0 11
1.1 250 1.0 11 1.1 250 1.0 11
Board 5: 250 1.0 11 1.1 250 1.0 11
1.1 250 1.0 11 1.1 250 1.0 11
Board 6: 250 1.0 11 1.1 250 1.0 11
1.1 250 1.0 11 1.1 250 1.0 11
Board 7: 250 1.0 11 1.1 250 1.0 11
1.1 250 1.0 11 1.1 250 1.0 11
Board 8: 250 1.0 11 1.1 250 1.0 11
1.1 250 1.0 11 1.1 250 1.0 11
Board 9: 250 1.0 11 1.1 250 1.0 11
1.1 250 1.0 11 1.1 250 1.0 11
Board10: 250 1.0 11 1.1 250 1.0 11
1.1 250 1.0 11 1.1 250 1.0 11
Board11: 250 1.0 11 1.1 250 1.0 11
1.1 250 1.0 11 1.1 250 1.0 11
Board12: 250 1.0 11 1.1 250 1.0 11
1.1 250 1.0 11 1.1 250 1.0 11
Board13: 250 1.0 11 1.1 250 1.0 11
1.1 250 1.0 11 1.1 250 1.0 11
Board14: 250 1.0 11 1.1 250 1.0 11
1.1 250 1.0 11 1.1 250 1.0 11
Board15: 250 1.0 11 1.1 250 1.0 11
1.1 250 1.0 11 1.1 250 1.0 11
Memory Units: Size
0: MB 1: MB 2: MB 3: MB
-----
Board 0 256 256 256 256
Board 1 256 256 256 256
Board 2 256 256 256 256
Board 3 256 256 256 256
Board 4 256 256 256 256
Board 5 256 256 256 256
Board 6 256 256 256 256
Board 7 256 256 256 256
Board 8 256 256 256 256
Board 9 256 256 256 256
Board10 256 256 256 256
Board11 256 256 256 256
Board12 256 256 256 256
Board13 256 256 256 256
Board14 256 256 256 256
Board15 256 256 256 256
=====IO
Cards=====
Board 0, SBus0:
SBus clock frequency: 25 MHz
0: QLGC,isp/sd(block) 'QLGC,ISP1000'
1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board 0, SBus1:
SBus clock frequency: 25 MHz
0: QLGC,isp/sd(block) 'QLGC,ISP1000'
1: qec/qe(network) 'SUNW,595-3198'

```

Hardware Configuration:

```

Sat Mar 8 09:34:42 PST 1997
System Configuration: Sun Microsystems sun4u SUNW,Ultra-
Enterprise-10000
System clock frequency: 83 MHz

```

```

Board 1, SBus0:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board 1, SBus1:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board 2, SBus0:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board 2, SBus1:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board 3, SBus0:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board 3, SBus1:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board 4, SBus0:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board 4, SBus1:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board 5, SBus0:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board 5, SBus1:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board 6, SBus0:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board 6, SBus1:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board 7, SBus0:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board 7, SBus1:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board 8, SBus0:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board 8, SBus1:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board 9, SBus0:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board 9, SBus1:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board10, SBus0:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board10, SBus1:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board11, SBus0:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board11, SBus1:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board12, SBus0:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board12, SBus1:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board13, SBus0:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board13, SBus1:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board14, SBus0:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'
Board14, SBus1:
  SBus clock frequency: 25 MHz
  0: QLGC,isp/sd(block) 'QLGC,ISP1000'
  1: QLGC,isp/sd(block) 'QLGC,ISP1000'

```

For diagnostic information,
see /var/opt/SUNWssp/adm/sael/messages on the SSP.
Searching for disks...done

AVAILABLE DISK SELECTIONS:

```

0. c0t0d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
  /sbus@40,0/QLGC,isp@0,10000/sd@0,0
1. c0t1d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
  /sbus@40,0/QLGC,isp@0,10000/sd@1,0
2. c1t0d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
  /sbus@40,0/QLGC,isp@1,10000/sd@0,0
3. c1t1d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
  /sbus@40,0/QLGC,isp@1,10000/sd@1,0
4. c1t2d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
  /sbus@40,0/QLGC,isp@1,10000/sd@2,0
5. c1t3d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
  /sbus@40,0/QLGC,isp@1,10000/sd@3,0
6. c1t4d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
  /sbus@40,0/QLGC,isp@1,10000/sd@4,0
7. c1t5d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
  /sbus@40,0/QLGC,isp@1,10000/sd@5,0
8. c1t6d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
  /sbus@40,0/QLGC,isp@1,10000/sd@6,0
9. c2t0d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
  /sbus@41,0/QLGC,isp@0,10000/sd@0,0
10. c2t1d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
  /sbus@41,0/QLGC,isp@0,10000/sd@1,0
11. c2t2d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
  /sbus@41,0/QLGC,isp@0,10000/sd@2,0
12. c2t3d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
  /sbus@41,0/QLGC,isp@0,10000/sd@3,0
13. c2t4d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
  /sbus@41,0/QLGC,isp@0,10000/sd@4,0
14. c2t5d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
  /sbus@41,0/QLGC,isp@0,10000/sd@5,0
15. c2t6d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
  /sbus@41,0/QLGC,isp@0,10000/sd@6,0
16. c3t0d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
  /sbus@44,0/QLGC,isp@0,10000/sd@0,0
17. c3t1d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
  /sbus@44,0/QLGC,isp@0,10000/sd@1,0
18. c3t2d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
  /sbus@44,0/QLGC,isp@0,10000/sd@2,0
19. c3t3d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
  /sbus@44,0/QLGC,isp@0,10000/sd@3,0
20. c3t4d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
  /sbus@44,0/QLGC,isp@0,10000/sd@4,0
21. c3t5d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
  /sbus@44,0/QLGC,isp@0,10000/sd@5,0
22. c3t6d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
  /sbus@44,0/QLGC,isp@0,10000/sd@6,0
23. c4t0d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>

```



```

    /sbus@78,0/QLGC,isp@0,10000/sd@3,0
356. c51t4d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@78,0/QLGC,isp@0,10000/sd@4,0
357. c51t5d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@78,0/QLGC,isp@0,10000/sd@5,0
358. c51t6d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@78,0/QLGC,isp@0,10000/sd@6,0
359. c52t0d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@78,0/QLGC,isp@1,10000/sd@0,0
360. c52t1d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@78,0/QLGC,isp@1,10000/sd@1,0
361. c52t2d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@78,0/QLGC,isp@1,10000/sd@2,0
362. c52t3d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@78,0/QLGC,isp@1,10000/sd@3,0
363. c52t4d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@78,0/QLGC,isp@1,10000/sd@4,0
364. c52t5d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@78,0/QLGC,isp@1,10000/sd@5,0
365. c52t6d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@78,0/QLGC,isp@1,10000/sd@6,0
366. c53t0d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@79,0/QLGC,isp@0,10000/sd@0,0
367. c53t1d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@79,0/QLGC,isp@0,10000/sd@1,0
368. c53t2d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@79,0/QLGC,isp@0,10000/sd@2,0
369. c53t3d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@79,0/QLGC,isp@0,10000/sd@3,0
370. c53t4d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@79,0/QLGC,isp@0,10000/sd@4,0
371. c53t5d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@79,0/QLGC,isp@0,10000/sd@5,0
372. c53t6d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@79,0/QLGC,isp@0,10000/sd@6,0
373. c54t0d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@79,0/QLGC,isp@1,10000/sd@0,0
374. c54t1d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@79,0/QLGC,isp@1,10000/sd@1,0
375. c54t2d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@79,0/QLGC,isp@1,10000/sd@2,0
376. c54t3d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@79,0/QLGC,isp@1,10000/sd@3,0
377. c54t4d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@79,0/QLGC,isp@1,10000/sd@4,0
378. c54t5d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@79,0/QLGC,isp@1,10000/sd@5,0
379. c54t6d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@79,0/QLGC,isp@1,10000/sd@6,0
380. c55t0d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@7c,0/QLGC,isp@0,10000/sd@0,0
381. c55t1d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@7c,0/QLGC,isp@0,10000/sd@1,0
382. c55t2d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@7c,0/QLGC,isp@0,10000/sd@2,0
383. c55t3d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@7c,0/QLGC,isp@0,10000/sd@3,0
384. c55t4d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@7c,0/QLGC,isp@0,10000/sd@4,0
385. c55t6d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@7c,0/QLGC,isp@0,10000/sd@6,0
386. c56t0d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@7c,0/QLGC,isp@1,10000/sd@0,0
387. c56t1d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@7c,0/QLGC,isp@1,10000/sd@1,0
388. c56t2d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@7c,0/QLGC,isp@1,10000/sd@2,0
389. c56t3d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@7c,0/QLGC,isp@1,10000/sd@3,0
390. c56t4d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@7c,0/QLGC,isp@1,10000/sd@4,0
391. c56t5d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@7c,0/QLGC,isp@1,10000/sd@5,0
392. c56t6d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@7c,0/QLGC,isp@1,10000/sd@6,0
393. c57t0d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@7d,0/QLGC,isp@0,10000/sd@0,0
394. c57t1d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@7d,0/QLGC,isp@0,10000/sd@1,0
395. c57t2d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@7d,0/QLGC,isp@0,10000/sd@2,0
396. c57t3d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@7d,0/QLGC,isp@0,10000/sd@3,0

```

```

397. c57t4d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@7d,0/QLGC,isp@0,10000/sd@4,0
398. c57t5d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@7d,0/QLGC,isp@0,10000/sd@5,0
399. c57t6d0 <SUN4.2G cyl 3974 alt 2 hd 19 sec 111>
    /sbus@7d,0/QLGC,isp@0,10000/sd@6,0
400. c58t0d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6c,0/QLGC,isp@0,10000/sd@0,0
401. c58t1d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6c,0/QLGC,isp@0,10000/sd@1,0
402. c58t2d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6c,0/QLGC,isp@0,10000/sd@2,0
403. c58t3d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6c,0/QLGC,isp@0,10000/sd@3,0
404. c58t4d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6c,0/QLGC,isp@0,10000/sd@4,0
405. c58t5d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6c,0/QLGC,isp@0,10000/sd@5,0
406. c58t6d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6c,0/QLGC,isp@0,10000/sd@6,0
407. c59t0d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6c,0/QLGC,isp@1,10000/sd@0,0
408. c59t1d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6c,0/QLGC,isp@1,10000/sd@1,0
409. c59t2d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6c,0/QLGC,isp@1,10000/sd@2,0
410. c59t3d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6c,0/QLGC,isp@1,10000/sd@3,0
411. c59t4d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6c,0/QLGC,isp@1,10000/sd@4,0
412. c59t5d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6c,0/QLGC,isp@1,10000/sd@5,0
413. c59t6d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6c,0/QLGC,isp@1,10000/sd@6,0
414. c60t0d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6d,0/QLGC,isp@0,10000/sd@0,0
415. c60t1d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6d,0/QLGC,isp@0,10000/sd@1,0
416. c60t2d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6d,0/QLGC,isp@0,10000/sd@2,0
417. c60t3d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6d,0/QLGC,isp@0,10000/sd@3,0
418. c60t4d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6d,0/QLGC,isp@0,10000/sd@4,0
419. c60t5d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6d,0/QLGC,isp@0,10000/sd@5,0
420. c60t6d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6d,0/QLGC,isp@0,10000/sd@6,0
421. c61t0d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6d,0/QLGC,isp@1,10000/sd@0,0
422. c61t1d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6d,0/QLGC,isp@1,10000/sd@1,0
423. c61t2d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6d,0/QLGC,isp@1,10000/sd@2,0
424. c61t3d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6d,0/QLGC,isp@1,10000/sd@3,0
425. c61t4d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6d,0/QLGC,isp@1,10000/sd@4,0
426. c61t5d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6d,0/QLGC,isp@1,10000/sd@5,0
427. c61t6d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
    /sbus@6d,0/QLGC,isp@1,10000/sd@6,0

```

```

Specify disk (enter its number): selecting c0t0d0
[disk formatted]

```

FORMAT MENU:

```

disk          - select a disk
type          - select (define) a disk type
partition    - select (define) a partition table
current      - describe the current disk
format       - format and analyze the disk
repair       - repair a defective sector
label        - write label to the disk
analyze      - surface analysis
defect       - defect list management
backup       - search for backup labels
verify       - read and display labels
save         - save new disk/partition definitions
inquiry      - show vendor, product and revision
volname     - set 8-character volume name
quit

```

```
format>
```

Appendix B. Programs and Scripts

```

-----
load_go.sh
-----

#!/usr/bin/ksh

export LDOUT=$PWD/ldeventlog

echo starting everything `date` > $LDOUT

cat tpcd_database_B.sh >> $LDOUT 2>&1
tpcd_database_B.sh >> $LDOUT 2>&1

echo timed load phase start `date` >> $LDOUT

cat tpcd_schema_B.sh >> $LDOUT 2>&1
tpcd_schema_B.sh >> $LDOUT 2>&1

cat tpcd_load_B.sh >> $LDOUT 2>&1
tpcd_load_B.sh >> $LDOUT 2>&1

cat alter_par_p.sh >>$LDOUT 2>&1
alter_par_p.sh 128 >>$LDOUT 2>&1

cat tpcd_analyze_tbl.sh >> $LDOUT 2>&1
tpcd_analyze_tbl.sh >> $LDOUT 2>&1

cat tpcd_idx_B.sh >> $LDOUT 2>&1
tpcd_idx_B.sh >> $LDOUT 2>&1

cat tpcd_analyze_idx.sh >> $LDOUT 2>&1
tpcd_analyze_idx.sh >> $LDOUT 2>&1

cat alter_par_p.sh >>$LDOUT 2>&1
alter_par_p.sh 96 >>$LDOUT 2>&1

echo timed load phase complete `date` >> $LDOUT

rm dbcheck.lst dbtable.lst

echo start dbcheck and dbtables `date` >> $LDOUT
$qlplus $PD/$PD < $ORACLE_HOME/tpcd/audit/dbcheck.sql >
/dev/null 2>&1
$qlplus $PD/$PD < $ORACLE_HOME/tpcd/audit/dbtables.sql >
/dev/null 2>&1
mv dbcheck.lst rdbcheckt
mv dbtable.lst rdbtablest
echo end dbcheck and dbtables `date` >> $LDOUT

echo done with everything `date` >> $LDOUT

-----
tpcd_database_B.sh
-----
#!/bin/sh

echo "Started DB Build at `date` "

STARTDIR=$PWD
cd /export/tpcd/sf300/create_db

cat createdb.sh
createdb.sh

cat add_all_rbs.sh
add_all_rbs.sh

svrmgrl << EOF
connect internal
shutdown
startup pfile=/export/tpcd/sf300/create_db/inittpcd.ora
exit;
EOF

cat add_all_ts.sh
add_all_ts.sh

cd $STARTDIR

echo "Completed DB Build at `date` "

-----
createdb.sh
-----
#!/bin/sh

echo "Start createdb.sh at `date`"

mkdir $BENCH_HOME/dbs
mkdir $BENCH_HOME/rdbms
mkdir $BENCH_HOME/rdbms/log

rm $BENCH_HOME/dbs/*tpcd*
rm $BENCH_HOME/rdbms/log/*sf300*.log
rm $BENCH_HOME/rdbms/log/*sf300*.trc

cat create_db.sql
svrmgrl << EOF
@create_db
exit;
EOF

rm -rf catalog.out > /dev/null
echo Oracle Catalog output stored in $PWD/catalog.out

cat create_db_cat.sql
svrmgrl > catalog.out << EOF
@create_db_cat
exit;
EOF

echo "End database catalog at `date`"

echo "Starting tablespace creates at `date`"

cat cre_ts.sh
cre_ts.sh

echo "Starting create user at `date`"

cat cre_user.sql
svrmgrl << EOF
connect internal
@cre_user
exit;
EOF

echo "Done with basic database creation at `date`"

-----
create_db.sql
-----
REM * Set terminal output and command echoing on; log
output of this script.
REM *
set termout on
set echo on
spool create_db.lst

REM * Start the instance (ORACLE_SID here must be set to
<db_name>).
REM *
connect internal
startup nomount pfile=inittpcd_0.ora

REM * Create the tpcd database.
REM * SYSTEM tablespace configuration guidelines:
REM *   General-Purpose ORACLE RDBMS      5Mb
REM *   Additional dictionary for applications10-50Mb
REM * Redo Log File configuration guidelines:
REM *   Use 3+ redo log files to relieve ``cannot allocate
new log... waits.
REM *   Use ~100Kb per redo log file per connection to
reduce checkpoints.
REM *
create database sf300
maxdatafiles 1022
maxlogfiles 32
maxlogmembers 5
maxinstances 10
character set "US7ASCII"

```

```

controlfile reuse
datafile
'/dev/vx/rdisk/sf300_sys_0' size 56288K
logfile
group REUSE, 1 ( '/dev/vx/rdisk/valid/sf300_redo_0') size 1999M
group REUSE, 2 ( '/dev/vx/rdisk/valid/sf300_redo_1') size 1999M
group REUSE, 3 ( '/dev/vx/rdisk/valid/sf300_redo_2') size 1999M

host sleep 30
REM * Create additional rollback segment in SYSTEM before
creating tablespace.
REM *
create rollback segment r0 tablespace system
storage (initial 16k next 16k minextents 2 maxextents 10);

REM * Activate the additional rollback segment.
REM *
alter rollback segment r0 online;

REM * Create a tablespace for rollback segments.
REM * Rollback segment configuration guidelines:
REM * 1 rollback segments for every 4 concurrent
xactions.
REM * No more than 50 rollback segments.
REM * All rollback segments the same size.
REM * Between 20 and 40 homogeneously-sized extents per
rollback segment.
REM *
create tablespace rbs datafile
'/dev/vx/rdisk/sf300_rbs_0' size 272608K
default storage (
initial 10M
next 10M
maxextents 121
);

REM * Create rollback segments.
REM *
create rollback segment r01 tablespace rbs
storage (initial 512k next 512k );
create rollback segment r02 tablespace rbs
storage (initial 512k next 512k );
create rollback segment r03 tablespace rbs
storage (initial 512k next 512k );
create rollback segment r04 tablespace rbs
storage (initial 512k next 512k );
create rollback segment r05 tablespace rbs
storage (initial 512k next 512k );
create rollback segment r06 tablespace rbs
storage (initial 512k next 512k );

REM * Alter new rollback segments online and r0 offline.
REM *
alter rollback segment r01 online;
alter rollback segment r02 online;
alter rollback segment r03 online;
alter rollback segment r04 online;
alter rollback segment r05 online;
alter rollback segment r06 online;
alter rollback segment r0 offline;

REM * Create a tablespace for rollback segments.
REM * Rollback segment configuration guidelines:
REM * 1 rollback segments for every 4 concurrent
xactions.
REM * No more than 50 rollback segments.
REM * All rollback segments the same size.
REM * Between 20 and 40 homogeneously-sized extents per
rollback segment.
REM *
create tablespace rbs1 datafile
'/dev/vx/rdisk/sf300_rbs_1' size 545216K
default storage (
initial 10M
next 10M
maxextents 121
);

create tablespace rbs2 datafile
'/dev/vx/rdisk/sf300_rbs_2' size 545216K
default storage (
initial 10M
next 10M
maxextents 121
);

create tablespace rbs3 datafile
'/dev/vx/rdisk/sf300_rbs_3' size 545216K
default storage (
initial 10M
next 10M
maxextents 121
);

REM * Create a tablespace for database user activity.
REM *
create tablespace user_ts datafile
'/dev/vx/rdisk/sf300_user_0' size 542944K
default storage (
initial 1M
next 10M
minextents 1
maxextents 100
pctincrease 0);

spool off

-----
create_db_cat.sql
-----
REM * Set terminal output and command echoing on; log
output of this script.
REM *
set termout on
set echo on
spool create_db_cat.lst
connect internal

REM * Create dictionary views, export utility views, and
pl/sql script.
REM * On 7.0.12 catalog.sql errors if run as an
init_sql_files.
REM *
@${ORACLE_HOME}/rdbms/admin/catalog.sql
set termout off
REM * Script to create procedures and function objects.
@${ORACLE_HOME}/rdbms/admin/catproc.sql
REM * Script to allow snapshot creation
@${ORACLE_HOME}/rdbms/admin/catsnap.sql
REM * Script to allow snapshot administration
@${ORACLE_HOME}/rdbms/admin/dbmssnap.sql
REM * Script to allow parallel administration
@${ORACLE_HOME}/rdbms/admin/catparr.sql
set termout on

REM * For each DBA user, run DBA synonyms SQL script.
Don't forget that EACH
REM * DBA USER created in the future needs dba_syn.sql run
from its account.
REM *
connect system/manager
@${ORACLE_HOME}/rdbms/admin/catdbsyn.sql
@${ORACLE_HOME}/sqlplus/admin/pubpld

spool off

-----
cre_ts.sh
-----
#!/bin/sh

for i in li ps ind su o pt c temp
do
init=64K
next=1M
case $i in
li) n=lineitem_ts
s=1964368K
;;
ps) n=partsupp_ts
s=645664K
;;
ind) n=indx_ts
s=4532960K
;;
o) n=orders_ts
s=922592K
;;
);

```

```

pt) n=parts_ts
    s=152160K
    ;;
c) n=customer_ts
    s=137056K
    ;;
su) n=supplier_ts
    s=9184K
    ;;
temp) n=temp
      s=1964368K
      next=200M
      ;;
      esac

      echo "Creating tablespace $n at `date`"
      (
      svrmgrl << EOF
      connect internal

      REM * Create a tablespace for database activity.
      REM *
      spool $n.lst
      drop tablespace ${n};
      create tablespace ${n} datafile
      \dev\vx\rsk\rootdg\sf300_${n}_0
      size $s
      default storage (
      initial $init
      next $next
      pctincrease 0);
      spool off;
      exit;
      EOF
      echo "Done creating tablespace ${n} at `date`"
      ) &
done

wait

-----
cre_user.sql
-----
spool cre_user.lst;
set echo on;
create user bench identified by bench
      default tablespace user_ts
      temporary tablespace temp
      quota unlimited on temp;

grant connect,resource to bench;
grant dba to bench;
connect bench/bench

@?/rdbms/admin/utlxplan.sql

spool off;
exit;

-----
add_all_rbs.sh
-----
#!/bin/sh

start=`date`

echo "starting adding rollback segments at $start"

cat add_roll.sh

add_roll.sh 1 t 1 50
add_roll.sh 2 u 1 50
add_roll.sh 3 v 1 50

stop=`date`
echo done adding rollback segments from $start to $stop

-----
add_roll.sh
-----
#!/bin/sh

TBSP=rbs$1
SEG_PRE=$2

i=$3
while [ $i -le $4 ]
do
(
svrmgrl << EOF
connect internal

create rollback segment $SEG_PRE$i tablespace $TBSP
storage (initial 512k next 512k minextents 10 );
alter rollback segment $SEG_PRE$i online;
exit;
EOF
)

i=`expr $i + 1`
done

-----
add_all_ts.sh
-----
#!/bin/sh

start=`date`

echo "starting adding files to tablespaces at $start"

cat add_ts.sh
add_ts.sh lineitem_ts 128 li 1964368K 300
add_ts.sh partsupp_ts 64 ps 645664K 300
add_ts.sh indx_ts 64 ind 4532960K 300
add_ts.sh supplier_ts 64 su 9184K 300
add_ts.sh orders_ts 64 o 922592K 300
add_ts.sh parts_ts 64 pt 152160K 300
add_ts.sh customer_ts 64 c 137056K 300
add_ts.sh temp 128 temp 1964368K 300

wait

stop=`date`
echo done adding files to tablespaces from $start to $stop

-----
add_ts.sh
-----
#!/bin/sh

name=$1
num=$2
abbrev=$3
size=$4
sf=$5

start=`date`
echo start $name at $start

i=1
while [ $i -lt $num ]
do
(
svrmgrl << EOF
connect internal

alter tablespace ${name}
add datafile \dev\vx\rsk\rootdg\sf${sf}_${abbrev}_${i}'
size $size;
exit;
EOF
) &

i=`expr $i + 1`
done

wait
stop=`date`
echo completed $name from $start to $stop

-----
tpcd_schema_B.sh
-----
#!/bin/sh

echo "Started DB Schema at `date` "

```

```

for TABLE_NAME in lineitem orders partsupp customer parts
supplier nation region
do
    cat $BENCH_HOME/tables/$TABLE_NAME.sql

    sqlplus -s $PD/$PD <<EOF
    @$BENCH_HOME/tables/$TABLE_NAME.sql
    exit
EOF
done

echo "Completed DB Schema at `date` "

-----
lineitem.sql
-----
drop table lineitem;

CREATE TABLE LINEITEM (
    l_shipdate DATE CONSTRAINT nn_l_shipdate NOT NULL,
    l_orderkey NUMBER(10) CONSTRAINT nn_l_orderkey NOT NULL,
    l_discount NUMBER(12,2) CONSTRAINT nn_l_discount NOT
NULL,
    l_extendedprice NUMBER(12,2) CONSTRAINT
nn_l_extendedprice NOT NULL,
    l_suppkey NUMBER(10) CONSTRAINT nn_l_suppkey NOT NULL,
    l_quantity NUMBER(12,2) CONSTRAINT nn_l_quantity NOT
NULL,
    l_returnflag CHAR(1) CONSTRAINT nn_l_returnflag NOT
NULL,
    l_partkey NUMBER(10) CONSTRAINT nn_l_partkey NOT NULL,
    l_linestatus CHAR(1) CONSTRAINT nn_l_linestatus NOT
NULL,
    l_tax NUMBER(12,2) CONSTRAINT nn_l_tax NOT NULL,
    l_commitdate DATE CONSTRAINT nn_l_commitdate NOT NULL,
    l_receiptdate DATE CONSTRAINT nn_l_receiptdate NOT NULL,
    l_shipmode VARCHAR2(10) CONSTRAINT nn_l_shipmode NOT
NULL,
    l_linenum NUMBER(10) CONSTRAINT nn_l_linenum NOT
NULL,
    l_shipinstruct VARCHAR2(25) CONSTRAINT
nn_l_shipinstruct NOT NULL,
    l_comment VARCHAR2(44) CONSTRAINT nn_l_comment NOT NULL
)
pctfree 3 pctused 96
intrans 4
tablespace lineitem_ts
storage (initial 64K
next 1658000K
minextents 1
maxextents unlimited
FREELISTS 9
FREELIST GROUPS 64
pctincrease 0);

-----
orders.sql
-----
drop table orders;

CREATE TABLE ORDERS
(
    o_orderdate DATE CONSTRAINT nn_o_orderdate NOT NULL,
    o_orderkey NUMBER(10) CONSTRAINT nn_o_orderkey NOT NULL,
    o_custkey NUMBER(10) CONSTRAINT nn_o_custkey NOT NULL,
    o_orderpriority VARCHAR2(15) CONSTRAINT
nn_o_orderpriority NOT NULL,
    o_shippriority NUMBER(10) CONSTRAINT nn_o_shippriority
NOT NULL,
    o_clerk VARCHAR2(15) CONSTRAINT nn_o_clerk NOT NULL,
    o_orderstatus CHAR(1) CONSTRAINT nn_o_orderstatus NOT
NULL,
    o_totalprice NUMBER(12,2) CONSTRAINT nn_o_totalprice NOT
NULL,
    o_comment VARCHAR2(79) CONSTRAINT nn_o_comment NOT NULL
)
pctfree 3 pctused 96
INTRANS 4
tablespace orders_ts
storage (initial 64K
next 921440K
MAXEXTENTS unlimited
FREELISTS 9
FREELIST GROUPS 64
pctincrease 0);

-----
partsupp.sql
-----

drop table partsupp;

CREATE TABLE PARTSUPP
(
    ps_partkey NUMBER(10) CONSTRAINT nn_ps_partkey NOT
NULL,
    ps_suppkey NUMBER(10) CONSTRAINT nn_ps_suppkey NOT
NULL,
    ps_supplycost NUMBER(12,2) CONSTRAINT nn_ps_supplycost
NOT NULL,
    ps_availqty NUMBER(10) CONSTRAINT nn_ps_availqty NOT
NULL,
    ps_comment VARCHAR2(199) CONSTRAINT nn_ps_comment NOT
NULL
)
pctfree 0 pctused 99
tablespace partsupp_ts
storage (initial 64K
next 644512K
minextents 1
maxextents unlimited
FREELIST GROUPS 64
pctincrease 0);

-----
customer.sql
-----
drop table customer;

CREATE TABLE CUSTOMER (
    c_custkey NUMBER(10) CONSTRAINT nn_c_custkey NOT
NULL,
    c_mktsegment CHAR(10) CONSTRAINT nn_c_mktsegment NOT
NULL,
    c_nationkey NUMBER(10) CONSTRAINT nn_c_nationkey NOT
NULL,
    c_name VARCHAR2(25) CONSTRAINT nn_c_name NOT NULL,
    c_address VARCHAR2(40) CONSTRAINT nn_c_address NOT
NULL,
    c_phone CHAR(15) CONSTRAINT nn_c_phone NOT NULL,
    c_acctbal NUMBER(12,2) CONSTRAINT nn_c_acctbal NOT
NULL,
    c_comment VARCHAR2(117) CONSTRAINT nn_c_comment NOT
NULL
)
PCTFREE 0
PCTUSED 99
tablespace customer_ts
storage (initial 64K
next 135904K
minextents 1
maxextents unlimited
FREELIST GROUPS 64
pctincrease 0);

-----
parts.sql
-----
drop table parts;

CREATE TABLE PARTS
(
    p_partkey NUMBER(10) CONSTRAINT nn_p_partkey NOT
NULL,
    p_type VARCHAR2(25) CONSTRAINT nn_p_type NOT NULL,
    p_size NUMBER(10) CONSTRAINT nn_p_size NOT NULL,
    p_brand CHAR(10) CONSTRAINT nn_p_brand NOT NULL,
    p_name VARCHAR2(55) CONSTRAINT nn_p_name NOT NULL,
    p_container CHAR(10) CONSTRAINT nn_p_container NOT NULL,
    p_mfgr CHAR(25) CONSTRAINT nn_p_mfgr NOT NULL,
    p_retailprice NUMBER(12,2) CONSTRAINT nn_p_retailprice
NOT NULL,
    p_comment VARCHAR2(23) CONSTRAINT nn_p_comment NOT NULL
)
pctfree 0 pctused 99
tablespace parts_ts
storage (initial 64K
next 151008K
minextents 1
MAXEXTENTS unlimited
FREELIST GROUPS 64
pctincrease 0);

-----
supplier.sql
-----
drop table supplier;

CREATE TABLE SUPPLIER

```

```

(
s_suppkey NUMBER(10) CONSTRAINT nn_s_suppkey NOT
NULL,
s_nationkey NUMBER(10) CONSTRAINT nn_s_nationkey NOT
NULL,
s_comment VARCHAR2(101) CONSTRAINT nn_s_comment NOT
NULL,
s_name CHAR(25) CONSTRAINT nn_s_name NOT NULL,
s_address VARCHAR2(40) CONSTRAINT nn_s_address NOT
NULL,
s_phone CHAR(15) CONSTRAINT nn_s_phone NOT NULL,
s_acctbal NUMBER(12,2) CONSTRAINT nn_s_acctbal NOT NULL
)
pctfree 0 pctused 99
tablespace supplier_ts
storage (initial 64K
next 8032K
minextents 1
maxextents unlimited
FREELIST GROUPS 64
pctincrease 0);

-----
nation.sql
-----
drop table nation;

CREATE TABLE NATION
(
n_nationkey NUMBER(10) CONSTRAINT nn_n_nationkey NOT
NULL,
n_name VARCHAR2(25) CONSTRAINT nn_n_name NOT NULL,
n_regionkey NUMBER(10) CONSTRAINT nn_n_regionkey NOT
NULL,
n_comment VARCHAR2(152) CONSTRAINT nn_n_comment NOT
NULL
)
pctfree 0 pctused 99
tablespace system CACHE;

-----
region.sql
-----
drop table region;

CREATE TABLE REGION
(
r_regionkey NUMBER(10) CONSTRAINT nn_r_regionkey NOT
NULL,
r_name CHAR(25) CONSTRAINT nn_r_name NOT NULL,
r_comment VARCHAR2(152) CONSTRAINT nn_r_comment NOT NULL
)
pctfree 0 pctused 99
tablespace system cache;

-----
tpcd_load_B.sh
-----
#!/bin/sh

echo "Started DB Load at `date` "

START_DIR=$PWD
cd /export/tpcd/sf300/dbgen

cat run_dbgen.sh
run_dbgen.sh all

cd $START_DIR
echo "Completed DB Load at `date` "

-----
run_dbgen.sh
-----
#!/bin/sh
#
# Script to create the database and to create and load the
# tables for the
# sf300 TPCD benchmark. Will do a parallel load of
# L,O,PS,C,P by invoking 10
# instances of dbgen 1.0 in separate subdirectories and
# directing each
# output to a named pipe. Only 1 dbspace per raw device
# is used, so each
# dbspace will contain data for each of above tables plus
# S.
#
# %W% %G%
#
# Should change to rm pipe before, and after the
# reload_multi_way functions
#
# Location of dbgen executable and data generation files
#
GDIR="/export/tpcd/dbgen.1.2.0"
WAYS=64

init_links()
{
#
# Set up links
#
if [ ! -f dists.dss ]; then
ln -s $GDIR/dists.dss
fi
if [ ! -f dbgen ]; then
ln -s $GDIR/dbgen
fi
}

DSS_SEED="$GDIR/seeds";export DSS_SEED

L_TABLE_NAME="lineitem"
O_TABLE_NAME="orders"
PS_TABLE_NAME="partsupp"
C_TABLE_NAME="customer"
P_TABLE_NAME="parts"
S_TABLE_NAME="supplier"
N_TABLE_NAME="nation"
R_TABLE_NAME="region"

L_DBGEN_TABLE_SWITCH="L"
O_DBGEN_TABLE_SWITCH="O"
PS_DBGEN_TABLE_SWITCH="S"
C_DBGEN_TABLE_SWITCH="C"
P_DBGEN_TABLE_SWITCH="P"
S_DBGEN_TABLE_SWITCH="S"
N_DBGEN_TABLE_SWITCH="n"
R_DBGEN_TABLE_SWITCH="r"

L_TABLE_DATA="lineitem.tbl"
O_TABLE_DATA="order.tbl"
PS_TABLE_DATA="partsupp.tbl"
C_TABLE_DATA="customer.tbl"
P_TABLE_DATA="part.tbl"
S_TABLE_DATA="supplier.tbl"
N_TABLE_DATA="nation.tbl"
R_TABLE_DATA="region.tbl"

reload_multi_way()
{
echo ""
echo "... Loading table $TABLE_NAME data at `date`"
echo ""

init_links

s=1
while [ $s -le $WAYS ]
do
rm -rf $TABLE_NAME.$s
mkdir $TABLE_NAME.$s
cd $TABLE_NAME.$s
init_links

/usr/sbin/mknod $TABLE_DATA.$s p

dbgen -T $DBGEN_TABLE_SWITCH -s 300 -S $s -C $WAYS &
time sqldr userid=bench/bench load=999999999 \
silent=feedback control=../$TABLE_NAME.ct1
data=$TABLE_DATA.$s \
parallel=true direct=true log=$TABLE_NAME.log &
cd ..

s=`expr $s + 1`
done
wait

```

```

echo ""
echo "... Done loading table $TABLE_NAME data at `date`"
echo ""
}

reload_lway()
{
init_links

echo ""
echo "... Loading table $TABLE_NAME data at `date`"
echo ""

rm -f $TABLE_DATA
/usr/sbin/mknod $TABLE_DATA p
dbgen -T $DBGEN_TABLE_SWITCH -s 300 &

time sqlldr userid=bench/bench load=999999999
silent=feedback \
control=$TABLE_NAME.ctl data=$TABLE_DATA
log=$TABLE_NAME.log &

wait
echo ""
echo "... Done loading table $TABLE_NAME data at `date`"
echo ""
}

load_one()
{
WAYS=64

case $1 in
region)

TABLE_NAME="$R_TABLE_NAME"
TABLE_DATA="$R_TABLE_DATA"
DBGEN_TABLE_SWITCH="$R_DBGEN_TABLE_SWITCH"

reload_lway
;;

nation)

TABLE_NAME="$N_TABLE_NAME"
TABLE_DATA="$N_TABLE_DATA"
DBGEN_TABLE_SWITCH="$N_DBGEN_TABLE_SWITCH"

reload_lway
;;

supplier)

TABLE_NAME="$S_TABLE_NAME"
TABLE_DATA="$S_TABLE_DATA"
DBGEN_TABLE_SWITCH="$S_DBGEN_TABLE_SWITCH"

WAYS=32
reload_multi_way
;;

parts)

TABLE_NAME="$P_TABLE_NAME"
TABLE_DATA="$P_TABLE_DATA"
DBGEN_TABLE_SWITCH="$P_DBGEN_TABLE_SWITCH"

reload_multi_way
;;

customer)

TABLE_NAME="$C_TABLE_NAME"
TABLE_DATA="$C_TABLE_DATA"
DBGEN_TABLE_SWITCH="$C_DBGEN_TABLE_SWITCH"

reload_multi_way
;;

partsupp)

TABLE_NAME="$PS_TABLE_NAME"
TABLE_DATA="$PS_TABLE_DATA"
DBGEN_TABLE_SWITCH="$PS_DBGEN_TABLE_SWITCH"

reload_multi_way
;;

orders)

TABLE_NAME="$O_TABLE_NAME"
TABLE_DATA="$O_TABLE_DATA"
DBGEN_TABLE_SWITCH="$O_DBGEN_TABLE_SWITCH"

reload_multi_way
;;

lineitem)

TABLE_NAME="$L_TABLE_NAME"
TABLE_DATA="$L_TABLE_DATA"
DBGEN_TABLE_SWITCH="$L_DBGEN_TABLE_SWITCH"

reload_multi_way
;;

*)
echo "Table $1 not recognised"
;;
esac
}

while [ $# -ne 0 ]
do
case "$1" in
all)
load_one lineitem
load_one nation
load_one orders
load_one partsupp
load_one parts
load_one region
load_one customer
load_one supplier
;;
*)
load_one $1
;;
esac
shift
done

-----
alter_par_p.sh
-----
#!/bin/sh
time sqlplus $PD/$PD << EOF
alter table lineitem parallel (degree $1);
alter table orders parallel (degree $1);
alter table parts parallel (degree $1);
alter table partsupp parallel (degree $1);
alter table supplier parallel (degree $1);
alter table customer parallel (degree $1);
alter table region parallel (degree 1) cache;
alter table nation parallel (degree 1) cache;
exit
EOF

-----
tpcd_analyze_tbl.sh
-----
#!/sbin/sh
#
#Copyright (c) 1994 by Sun Microsystems, Inc.
#
# %Z%M%I%E%SMI
#
#Analyze TPC-D database and compute optimizer statistics
in parallel
#This must be done or performance could be negatively
impacted!
#
# 10-Mar-1995 fpolizoPorted to TPC-D
# 30-Mar-1995 fpolizoRewritten for parallel execution.

```

```

ECHO=/bin/echo

$ECHO "Started tbl Analyze at `date`"

#####
# Functions
#####

#####
# estimate_table_stats_bg(table [, n rows ]) estimate
Oracle optimizer
# statistics for the named table as well as any indexes
defined for
# that table as a background process.
# {i}p., does "analyze table $1 estimate statistics $2
rows"
#####

estimate_table_stats_bg()
{
    if [ $# -le 1 ]
    then
    OPTIONS=""
    else
    OPTIONS="sample $2 rows"
    fi
    svrmgrl <<! &

CONNECT $PD/$PD
SET TIMING ON;

analyze table $1 estimate statistics ${OPTIONS};
!
}

#####
# table_exists(table) determines if the named table
exists
# returns 1 if it does, otherwise returns 0
#####
table_exists()
{
TBL="'$1'"
TC=`sqlplus -s $PD/$PD <<!
set hea off;
select count(*) from user_tables where table_name = $TBL ;
!`
TC=${ECHO} $TC | sed 's/W.*\./\.'/`
if [ ! -z "$TC" ] && [ $TC -eq 1 ]
then
    ${ECHO} 1
else
    ${ECHO} 0
fi
}

#####
# Mainline
#####

for table in lineitem orders partsupp parts customer
supplier nation region
do
    estimate_table_stats_bg $table 200000
done

echo doing tables
wait

$ECHO "Completed tbl Analyze at `date`"

-----
tpcd_idx_B.sh
-----
echo "Started Indices at `date` "

svrmgrl <<EOT
CONNECT $PD/$PD
ALTER TABLE lineitem STORAGE (NEXT 1M);
ALTER TABLE orders STORAGE (NEXT 1M);
EXIT
EOT

# Supplier Table Index

svrmgrl <<EOT
CONNECT $PD/$PD
SET TIMING ON;
DROP INDEX pk_s_suppkey;
CREATE INDEX pk_s_suppkey ON supplier (s_suppkey)
UNRECOVERABLE
TABLESPACE indx_ts
STORAGE (INITIAL 1M NEXT 1M
MAXEXTENTS unlimited
PCTINCREASE 0)
PCTFREE 0
PARALLEL (DEGREE DEFAULT)
;
EXIT
EOT

# Region Table Index

svrmgrl <<EOT
CONNECT $PD/$PD
SET TIMING ON;
DROP INDEX pk_r_regionkey;
CREATE UNIQUE INDEX pk_r_regionkey ON region (r_regionkey,
r_name)
UNRECOVERABLE
TABLESPACE indx_ts
STORAGE (INITIAL 8K NEXT 8K
PCTINCREASE 0)
PCTFREE 0
PARALLEL (DEGREE DEFAULT)
;
EXIT
EOT

# Partsupp Table Index

svrmgrl <<EOT
CONNECT $PD/$PD
SET TIMING ON;

DROP INDEX pk_ps_constraint;
ALTER TABLE partsupp DROP CONSTRAINT pk_ps_constraint;

CREATE UNIOUE INDEX pk_ps_constraint ON partsupp
(ps_partkey, ps_suppkey)
UNRECOVERABLE
TABLESPACE indx_ts
STORAGE (INITIAL 100M NEXT 100M
MAXEXTENTS unlimited
PCTINCREASE 0)
PCTFREE 0
PARALLEL (DEGREE DEFAULT)
;

ALTER TABLE partsupp ADD CONSTRAINT pk_ps_constraint
PRIMARY KEY (ps_partkey, ps_suppkey)
DISABLE;

ALTER TABLE partsupp ENABLE CONSTRAINT pk_ps_constraint;

EXIT
EOT

svrmgrl <<EOT
CONNECT $PD/$PD
SET TIMING ON;
DROP INDEX pk_ps_skey;
CREATE INDEX pk_ps_skey ON partsupp
(ps_suppkey,ps_partkey,ps_supplycost,ps_availqty)
UNRECOVERABLE
TABLESPACE indx_ts
STORAGE (INITIAL 100M NEXT 100M
MAXEXTENTS unlimited
PCTINCREASE 0)
PCTFREE 0
PARALLEL (DEGREE DEFAULT)
;
EXIT
EOT

# Order Table Index

```

```

svrmgrl <<EOT
CONNECT $PD/$PD
SET TIMING ON;
DROP INDEX pk_o_orderkey;
CREATE INDEX pk_o_orderkey ON orders (o_orderkey,
o_orderpriority)
  UNRECOVERABLE
  INITRANS 10
  TABLESPACE indx_ts
  STORAGE (INITIAL 100M NEXT 100M
           MAXEXTENTS unlimited
           PCTINCREASE 0
           FREELISTS 64
           FREELIST GROUPS 9)
  PCTFREE 2
  PARALLEL (DEGREE DEFAULT)
;
EXIT
EOT

svrmgrl <<EOT
CONNECT $PD/$PD
SET TIMING ON;
DROP INDEX pk_o_clokod;
CREATE INDEX pk_o_clokod ON orders (o_clerk,o_orderkey,
o_orderdate)
  UNRECOVERABLE
  INITRANS 10
  TABLESPACE indx_ts
  STORAGE (INITIAL 100M NEXT 100M
           MAXEXTENTS unlimited
           PCTINCREASE 0
           FREELISTS 64
           FREELIST GROUPS 9)
  PCTFREE 2
  PARALLEL (DEGREE DEFAULT)
;
EXIT
EOT

# Lineitem Table Index

svrmgrl <<EOT
CONNECT $PD/$PD
SET TIMING ON;

DROP INDEX pk_l_ored;
CREATE INDEX pk_l_ored ON lineitem (l_orderkey,
l_returnflag,l_extendedprice, l_discount)
  UNRECOVERABLE
  INITRANS 10
  TABLESPACE indx_ts
  STORAGE (INITIAL 100M NEXT 100M
           PCTINCREASE 0
           MAXEXTENTS unlimited
           FREELISTS 64
           FREELIST GROUPS 9)
  PCTFREE 2
  PARALLEL (DEGREE DEFAULT)
;

DROP INDEX pk_l_pgesod;
CREATE INDEX pk_l_pgesod ON lineitem (l_partkey,
quantity,l_extendedprice,
l_suppkey,l_orderkey,l_discount)
  UNRECOVERABLE
  INITRANS 10
  TABLESPACE indx_ts
  STORAGE (INITIAL 100M NEXT 100M
           PCTINCREASE 0
           MAXEXTENTS unlimited
           FREELISTS 64
           FREELIST GROUPS 9)
  PCTFREE 2
  PARALLEL (DEGREE DEFAULT)
;
EXIT
EOT

echo "Completed Indices at `date` "

-----
tpcd_analyze_idx.sh
-----

#!/sbin/sh
#
#Copyright (c) 1994 by Sun Microsystems, Inc.
#
# %Z%M%I%E%SMI
#
#Analyze TPC-D database and compute optimizer statistics
in parallel
#This must be done or performance could be negatively
impacted!
#
# 10-Mar-1995 fpolizoPorted to TPC-D
# 30-Mar-1995 fpolizoRewritten for parallel execution.

ECHO=/bin/echo

$ECHO "Started idx Analyze at `date`"

#####
# Functions
#####

#####
# estimate_index_stats_bg(index [, nrows ]) estimate
oracle optimizer-
# statistics for the named index as a background process.
# (i.e., does "analyze index $1 estimate statistics $2
rows")
#####
estimate_index_stats_bg()
{
    if [ $# -le 1 ]
    then
        OPTIONS=""
    else
        OPTIONS="sample $2 rows"
    fi
    sqldba lmode=y <<! &

CONNECT $PD/$PD
SET TIMING ON;

analyze index $1 estimate statistics ${OPTIONS};
!
}

#####
# table_exists(table) determines if the named table
exists
# returns 1 if it does, otherwise returns 0
#####
table_exists()
{
    TBL="$1"
    TC=`sqlplus -s $PD/$PD <<!
set hea off;
select count(*) from user_tables where table_name = $TBL ;
!`
    TC=`{ECHO} $TC | sed 's/w.*\./\/'`
    if [ ! -z "$TC" ] && [ $TC -eq 1 ]
    then
        ${ECHO} 1
    else
        ${ECHO} 0
    fi
}

#####
# Mainline
#####

for index in pk_s_suppkey pk_r_regionkey pk_ps_skey
pk_ps_constraint pk_o_orderkey pk_o_clokod pk_l_ored
pk_l_pgesod
do
    estimate_index_stats_bg $index 200000
done

echo doing indices
wait

$ECHO "Completed idx Analyze at `date`"

```

Appendix C. Query Text & Query Output

Begin Execution at Fri Mar 7 21:55:02 1997

-- using 1110820981 as a seed to the RNG
 -- Query 1 (Standard)

```
SELECT L_RETURNFLAG,
L_LINESTATUS,
SUM(L_QUANTITY) AS SUM_QTY,
SUM(L_EXTENDEDPRI) AS SUM_BASE_PRICE,
SUM(L_EXTENDEDPRI * (1 - L_DISCOUNT)) AS SUM_DISC_PRICE,
SUM(L_EXTENDEDPRI * (1 - L_DISCOUNT) * (1 + L_TAX)) AS
SUM_CHARGE,
AVG(L_QUANTITY) AS AVG_QTY,
AVG(L_EXTENDEDPRI) 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') -
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	11337858651.00	17001172526111.31	16151091002920.91	444620078.00	25.50	38237.53	0.05	1
N	F	295811034.00	443580663059.42	421394953116.85	11602354.00	25.50	38231.95	0.05	1
N	O	22592991808.00	33878109111519.36	32184261471537.51	33471645860190.55	25.50	38237.21	0.05	1
R	F	11337832829.00	17000876162091.54	16150835471439.83	16796849288672.39	25.50	38236.00	0.05	1

4 rows processed.
 Statement Processed in 2202.77 seconds.

Ended Executing this Query at Fri Mar 7 22:31:44 1997

Query Started at 857800502.18
 Query Ended at 857802704.95
 Query Processed in 2202.77 seconds.

-- Query 4 (Variant B)

Began Executing this Query at Fri Mar 7 22:31:44 1997

```
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('1993-11-01','YYYY-MM-DD')
AND O_ORDERDATE < ADD_MONTHS(TO_DATE('1993-11-01','YYYY-MM-DD'),3)
AND L_COMMITDATE < L_RECEIPTDATE)
GROUP BY O_ORDERPRIORITY
ORDER BY O_ORDERPRIORITY
```

O_ORDERPRIORITY	ORDER_COUNT
1-URGENT	3157773.00
2-HIGH	3160505.00
3-MEDIUM	3157813.00
4-NOT SPECIFIED	3157318.00
5-LOW	3158629.00

5 rows processed.
 Statement Processed in 721.85 seconds.

Ended Executing this Query at Fri Mar 7 22:43:46 1997

Query Started at 857802704.95
 Query Ended at 857803426.81
 Query Processed in 721.86 seconds.

-- Query 15 (Variant B)

Began Executing this Query at Fri Mar 7 22:43:46 1997

```
CREATE TABLE REVENUE0 (SUPPLIER_NO INTEGER, TOTAL_REVENUE
NUMBER(20,-2))
Statement Processed in 0.34 seconds.
```

```
INSERT INTO REVENUE0
SELECT L_SUPPKEY, SUM(L_EXTENDEDPRI*(1-L_DISCOUNT))
FROM LINEITEM
WHERE L_SHIPDATE >= TO_DATE('1993-11-01','YYYY-MM-DD')
AND L_SHIPDATE < ADD_MONTHS(TO_DATE('1993-11-01','YYYY-
MM-DD'),3)
GROUP BY L_SUPPKEY
Statement Processed in 555.53 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
2269800.00	Supplier#002269800	2269800.00	14-321-977-2964

1 row processed.
 Statement Processed in 185.74 seconds.

```
DROP TABLE REVENUE0
Statement Processed in 3.44 seconds.
Setting the number of rows to fetch to: 20
```

Ended Executing this Query at Fri Mar 7 22:56:11 1997

Query Started at 857803426.81
 Query Ended at 857804171.85
 Query Processed in 745.04 seconds.

-- Query 10 (Standard)

Began Executing this Query at Fri Mar 7 22:56:11 1997

```
SELECT
C_CUSTKEY,
C_NAME,
SUM(L_EXTENDEDPRI * (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('1993-05-01','YYYY-MM-DD')
AND O_ORDERDATE < ADD_MONTHS(TO_DATE('1993-05-01','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_ACCTBAL	C_NAME	REVENUE	C_PHONE
33093836.00 1029259.68	Customer#033093836 2110.51		
1029259.68	Customer#033093836 2110.51		
31830206.00 989589.84	Customer#031830206 2648.04		
31830206.00	Customer#031830206 2648.04		
30872299.00 319540.32	Customer#030872299 7570.71		
30872299.00	Customer#030872299 7570.71		
3803561.00 909146.08	Customer#003803561 1301.64		
3803561.00	Customer#003803561 1301.64		
30565061.00 308154.50	Customer#030565061 3584.16		
30565061.00	Customer#030565061 3584.16		
18591331.00 894772.93	Customer#018591331 9304.56		
18591331.00	Customer#018591331 9304.56		
42996527.00 894478.36	Customer#042996527 7804.23		
42996527.00	Customer#042996527 7804.23		
17260850.00 887183.54	Customer#017260850 620.77		
17260850.00	Customer#017260850 620.77		
28949455.00 899017.08	Customer#028949455 2488.03		
28949455.00	Customer#028949455 2488.03		
32438902.00 861217.19	Customer#032438902 3945.54		
32438902.00	Customer#032438902 3945.54		
32689490.00 860933.27	Customer#032689490 365.16		
32689490.00	Customer#032689490 365.16		
2864876.00 859835.76	Customer#002864876 5327.21		
2864876.00	Customer#002864876 5327.21		
528652.80	Customer#000528659 2496.87		
528652.80	Customer#000528659 2496.87		

xM03mw
 1711190CLiNkOMOhgrZjRPhBNQ1B66Rx26kA1Ox330S1267OR2Lhm5yLnhw
 PMR6P1Oyxwks8Ej54R2iY Lzm31y1jCh6ynRNL7
 34590917.00 Customer#034590917
 845187.17 1263.82
 INDONESIA
 19-801-644-3602 Bk3RB0QC3QzRyM2jwmSwi7h 0w62z
 kgL5S0m
 810101Mmkzjw01zhMM57ywnM3263OP07kxR6S1kjAC12B31R46j4myywBL
 4x5Rz2IRWRHyCCzRQIQ23gIA25440gJ1 WS5y257wnknInmQ
 1373338.00 Customer#001373338
 842608.95 7694.40
 GERMANY
 2CijMwQn2ByOOi0ih 17-893-902-7316
 S R4kin5Ams23hN6cARRln1MkMwgj722
 7m220Sag4hjn5iy70hiOLQCinsjn4M0CxlImMk
 8394052.00 Customer#009394052
 5598.36
 BRAZIL hgPMORPBC6nwm53w w6j61z
 12-777-173-2880
 106SyR2imjhkl1m5zMM5B0g32k10hg6OyyR5mlgmQyL50C6mlRM10gk

20 rows processed.
 Statement Processed in 953.77 seconds.

Ended Executing this Query at Fri Mar 7 23:12:05 1997

Query Started at 857804171.85
 Query Ended at 857805125.64
 Query Processed in 953.78 seconds.

-- Query 11 (Standard)

Began Executing this Query at Fri Mar 7 23:12:05 1997

```

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 = 'CANADA'
GROUP BY PS_PARTKEY
HAVING SUM(PS_SUPPLYCOST*PS_AVAILQTY) >
(SELECT
SUM(PS_SUPPLYCOST * PS_AVAILQTY) * 0.0000003333
FROM PARTSUPP, SUPPLIER, NATION
WHERE PS_SUPPKEY = S_SUPPKEY
AND S_NATIONKEY = N_NATIONKEY
AND N_NAME = 'CANADA' )
ORDER BY VALUE DESC

```

PS_PARTKEY	VALUE
12690766.00	25089836.98
54360772.00	24170541.84
27391709.00	23552073.90
10318841.00	22864341.53
9047359.00	22831283.41
10575527.00	22767163.06
16077745.00	22763435.70
26948603.00	22688248.92
183472.00	22640458.17
10752206.00	22316967.54
38778635.00	22254222.51
37442621.00	22206228.42
28322871.00	22189544.25
14667898.00	22158245.67
25657947.00	22128180.53
26309423.00	22117209.36
44646199.00	22023216.91
16913388.00	21965120.62
52748784.00	21904645.53
18621282.00	21704594.75
1214675.00	21693133.94
58743115.00	21515853.42
10146284.00	21506955.22
38220505.00	21489520.40
40360452.00	21335935.10
58834587.00	21298729.42
54286290.00	21292184.26
18903365.00	21229785.98
12149342.00	21222028.25
40183245.00	21191553.74

38615822.00	20936297.19	51640418.00	18500937.13
46497470.00	20887155.60	27315274.00	18490156.24
39307928.00	20704562.36	30762669.00	18487463.89
1202226.00	20634729.61	18032225.00	18480558.91
58740364.00	20611208.42	31801946.00	18476893.41
42836737.00	20566543.70	5323432.00	18471561.26
27329029.00	20508900.02	55102125.00	18465942.28
45930196.00	20438303.86	19916170.00	18445606.80
49588759.00	20432447.48	11485491.00	18432402.19
26934024.00	20414500.92	58727977.00	18431950.64
22939887.00	20413536.54	17364181.00	18424774.50
42252689.00	20409751.45	5927044.00	18419743.53
38376595.00	20364283.36	42810864.00	18413467.77
57333875.00	20286500.18	5557291.00	18398223.03
20091962.00	20230396.84	32318443.00	18327020.20
1581611.00	20136884.91	40801386.00	18323993.96
36894477.00	20080040.06	18345698.00	18310842.01
15539171.00	20042515.85	4370879.00	18292827.21
6848071.00	20021707.92	5338518.00	18288349.31
50364409.00	19966206.86	3028171.00	18285912.31
8030897.00	19963198.97	39076585.00	18284656.99
21808678.00	19962130.63	10306364.00	18266803.32
36401260.00	19958641.30	6985668.00	18266394.31
55711477.00	19935522.59	25932427.00	18264880.68
9478063.00	19907330.66	21440936.00	18239314.38
53580262.00	19881506.53	51773479.00	18228696.62
6025240.00	19879014.52	12852318.00	18214254.73
2017940.00	19790345.82	41943143.00	18213837.39
35478933.00	19732840.11	13246347.00	18207756.47
54769463.00	19705874.49	50071970.00	18197541.99
3148483.00	19527969.99	710749.00	18190533.92
45789967.00	19515392.48	11773011.00	18188555.88
27881736.00	19459171.58	36220836.00	18183746.13
4938803.00	19440201.32	14126845.00	18180090.12
15684455.00	19389147.16	59966876.00	18173815.78
34323858.00	19360310.19	9977016.00	18157769.36
51657133.00	19238480.31	35431251.00	18155003.80
39776547.00	19183720.02	47194438.00	18153937.12
45279758.00	19174832.42	23387770.00	18132828.10
58318362.00	19151983.88	24864250.00	18130971.29
48453545.00	19101590.00	8711481.00	18127021.12
18554203.00	19093303.87	10803251.00	18119770.74
35700796.00	19029650.68	15567443.00	18102288.40
2536759.00	19022929.55	33931002.00	18102018.21
52747955.00	19007461.34	52879334.00	18101111.31
19330085.00	18977600.04	35245109.00	18099340.38
32809788.00	18966327.58	55683088.00	18084732.38
14503316.00	18964872.15	32621065.00	18055567.94
48696258.00	18962911.54	51332088.00	18046280.48
6910872.00	18941796.17	37666235.00	18044072.14
12381180.00	18935876.32	44557245.00	18034496.70
42052277.00	18932998.24	46001322.00	18026851.21
38222609.00	18917060.88	11373147.00	18011267.50
29629716.00	18905672.93	27868073.00	18008435.31
34630043.00	18903069.76	44883069.00	18007810.14
10413589.00	18900880.91	56913320.00	18005349.27
33151063.00	18896937.96	38206195.00	18005228.78
50071319.00	18893537.19	15061287.00	17999936.12
21892042.00	18858405.47	12310634.00	17996161.83
57532578.00	18847247.07	38844916.00	17979264.10
43354415.00	18845094.95	47857789.00	17973837.60
45893487.00	18761798.60	20361941.00	17966485.27
46535789.00	18757072.36	10562325.00	17942519.84
49595114.00	18739757.55	58148667.00	17941325.84
7256673.00	18735923.54	19057265.00	17937529.10
52449472.00	18735309.56	17083189.00	17927992.23
37175293.00	18735026.13	26540492.00	17923570.95
8140953.00	18718687.00	4439352.00	17921048.01
28736851.00	18712736.97	22842756.00	17920859.16
10855843.00	18704577.06	32383358.00	17920056.14
47859152.00	18668210.77	16414716.00	17919704.46
30781653.00	18665131.93	49496733.00	17914404.93
9173996.00	18659090.40	25268603.00	17905955.58
28670474.00	18650210.98	41289525.00	17903951.07
6079117.00	18633844.75	32303987.00	17901478.94
11528190.00	18614781.29	34868740.00	17899864.36
22892178.00	18582461.60	56425645.00	17894366.20
35746348.00	18579604.18	44016439.00	17891098.20
22144272.00	18572401.84	47470325.00	17889946.44
42104906.00	18565978.75	10836248.00	17865447.58
41705460.00	18558443.76	44059540.00	17858931.02
45148399.00	18539432.85	7168013.00	17850273.43
32331979.00	18528598.79	10484706.00	17840540.37

```

46154679.00      17837863.62      S_ADDRESS      S_PHONE
19477647.00      17837429.31      S_COMMENT
27325168.00      17835544.86      9999.91      Supplier#000553792      FRANCE
57416753.00      17832501.66      9553791.00      Manufacturer#2
277308 rows processed.
Statement Processed in 370.07 seconds.

Ended Executing this Query at Fri Mar 7 23:18:15 1997

Query Started at 857805125.64
Query Ended at 857805495.71
Query Processed in 370.07 seconds.

-----
-- Query 6 (Standard)
-----
Began Executing this Query at Fri Mar 7 23:18:15 1997

SELECT
SUM(L_EXTENDEDPRI * L_DISCOUNT) AS REVENUE
FROM LINEITEM
WHERE L_SHIPDATE >= TO_DATE('1993-01-01','YYYY-MM-DD')
AND L_SHIPDATE < ADD_MONTHS(TO_DATE('1993-01-01','YYYY-MM-DD'),12)
AND L_DISCOUNT BETWEEN 0.07 - 0.01 AND 0.07 + 0.01
AND L_QUANTITY < 25

REVENUE
46951932986.53

1 row processed.
Statement Processed in 316.02 seconds.
Setting the number of rows to fetch to: 100

Ended Executing this Query at Fri Mar 7 23:23:31 1997

Query Started at 857805495.71
Query Ended at 857805811.73
Query Processed in 316.02 seconds.

-----
-- Query 2 (Variant A)
-----
Began Executing this Query at Fri Mar 7 23:23:31 1997

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 = 8
AND P_TYPE LIKE '%TIN'
AND S_NATIONKEY = N_NATIONKEY
AND N_REGIONKEY = R_REGIONKEY
AND R_NAME = 'EUROPE'
AND (P_PARTKEY,PS_SUPPLYCOST) IN
(SELECT PS_PARTKEY, MIN(PS_SUPPLYCOST)
FROM PARTSUPP, SUPPLIER, NATION, REGION
WHERE S_SUPPKEY = PS_SUPPKEY
AND S_NATIONKEY = N_NATIONKEY
AND N_REGIONKEY = R_REGIONKEY
AND R_NAME = 'EUROPE')
GROUP BY PS_PARTKEY
ORDER BY S_ACCTBAL DESC, N_NAME, S_NAME, P_PARTKEY

S_ACCTBAL      S_NAME      N_NAME
P_PARTKEY      P_MFGR

```

6161NM3g1CBNP13R52 21my20zPhyAwiRyk2Pnnh72n7S104NNNY Rgnk14gAF6PNMN7MPK	9996.60	Supplier#000964227	FRANCE
9998.55 Supplier#001402502 RUSSIA	48214210.00	Manufacturer#5	
56152447.00 Manufacturer#4	SSn6hMSQPjwyNyy6khCggz326zRj3h5zN2		16-557-307-8991
60iokggg75B0yk Qk 32-948-879-7250	wim656mwlkyhCxjmmSMgx4yOCM44		
821706n6Qy4N76PR1SOKP1S7PizNOLMx456j4N0jSMSPLM k7Mi71hn BB Ow378xMBMK Lom 5NwmM0hzgLB	9996.54 Supplier#001526971 ROMANIA		
9998.43 Supplier#002955761 UNITED KINGDOM	8276964.00 Manufacturer#3		29-597-488-1031
3705757.00 Manufacturer#2	RnR1xgSkS52hg0M2Ni5jn		
LAQS61RP5SwzyhJLM nMy 33-836-395-4406	ni05iGpP5CyRzV3nCL5m0L 6g70okjBkNkRSLj4N3 lyIz7Rkx6Q7xy31wAIR25644zIRAWlrxXB1CQnPh		
Ly641hnB2hiljRnB3xlPwgB0k5mw4SiBjzAQnnOw2w	9996.45 Supplier#000392828 GERMANY		
9998.27 Supplier#002310800 GERMANY	14642823.00 Manufacturer#2		
38310799.00 Manufacturer#5	152Q010m204kk4QR4myQillx jBP16R		17-965-114-2999
jAR414kAC2xlmLlCO0j4jlgMhkSxBkhh17 17-470-891-3647	mjP3Q6MP7N1C1OgB04m0m64BokCRR Mm6655mNvVj1A1Z0S71PwK5YSz018 OjJm13BhnNjy6mN7N1N6gn7A4P665047nRim		
h2SBPgnwnlnAC5khljSC1Az34QPA5BSSk0PxBjxS4	9996.45 Supplier#000795554 ROMANIA		
9998.22 Supplier#000270232 FRANCE	44295525.00 Manufacturer#2		29-305-384-2702
1020231.00 Manufacturer#3	i R0il67COhgCMzjzlw0w5Px1Q5kAR 37Sz		
Azzmg6AwP3kkx 16-519-317-1679	ROwPwMB34OmwhP3MnmN1M5BxNMwlywBnS5Q5B1L7xM01B3mxy0Mz7LS75i6		
iy12Q20smg64Aw7NyiOMNC3S0lyi1CyRhk16imgALNR7wngOwC Pfg7M1iSjn3gn65ngikmCSgzPyy	9996.37 Supplier#001182675 RUSSIA		
9998.22 Supplier#000270232 FRANCE	46932629.00 Manufacturer#3		
4020228.00 Manufacturer#2	3LyzyR 4kh1RRN6ngnlBNjQB xp77w M4		32-569-284-8959
Azzmg6AwP3kkx 16-519-317-1679	P5xS4Q27xLnR2iPi2y1nPQg3wx57CAQ1		
iy12Q20smg64Aw7NyiOMNC3S0lyi1CyRhk16imgALNR7wngOwC Pfg7M1iSjn3gn65ngikmCSgzPyy	9996.20 Supplier#001820308 RUSSIA		
9998.14 Supplier#001929948 UNITED KINGDOM	34070296.00 Manufacturer#3		
10179944.00 Manufacturer#1	P551g00L4hnBQQzQkmC7SL1BPYP5h		32-170-159-3993
x01RCRNhklmznMghCQS QO 33-726-837-5826	y512mx2Sm7wNmx15B5iBL60kBywCkllPi3g3yyjyg0klni10hr2gN65		
24h73M1672LM7wCz7Nih5lgkPwL4wLAMnS74RAi4yiBBlk1B	9996.15 Supplier#002303151 ROMANIA		
9998.04 Supplier#000265862 UNITED KINGDOM	20303150.00 Manufacturer#1		
4015858.00 Manufacturer#4	wCByj3mz hCgRCOM15iLx6iL xm17j5PB		29-243-350-2154
BMwP3yzLz0zAk5C6742C1CQB7212xwB 33-962-597-3811	115MmBj1BOiLmx5074Pim61gBjMm2Aw2m0543n5LMSRkxy1y2k33y2lQS7 wx7LzyPCSiY3iC1C1OCyN8		
R4C42iicM0OCA3AN43zN1PNNMga5NNA6hhmxQSSMAMhR4x770717wS0150 y1CmognjKRiywQ24BOK7BP5j7CwxMBA	9996.13 Supplier#002219753 FRANCE		
9998.04 Supplier#000265862 UNITED KINGDOM	40469739.00 Manufacturer#4		
34015828.00 Manufacturer#3	BQ54A1hj2R3ALBzB0QnwjP631A1i0iP NzziwLn		16-541-986-7992
BMwP3yzLz0zAk5C6742C1CQB7212xwB 33-962-597-3811	x36ABO7CBAOA01gyjw0z5P6k64k7yS55i7wBnlyBjlk		
R4C42iicM0OCA3AN43zN1PNNMga5NNA6hhmxQSSMAMhR4x770717wS0150 y1CmognjKRiywQ24BOK7BP5j7CwxMBA	9996.10 Supplier#002971475 GERMANY		
9997.99 Supplier#001979599 GERMANY	59971474.00 Manufacturer#5		
20729580.00 Manufacturer#1	iNh nkSSghkQhnBznM wSCL1MR6mx		17-835-750-3842
55Bn6All4kOnSLkh 3hikR2niiy6Q1lR 17-174-607-3972	45Mioz0RmBkCjigB5Ohz61Ln14CA1h4x6MPiPBjSP2Q		
y yzQLLN 0g5Q03PB1i5xMyMhQ5gjmABxwNLnN5kRPMBwvlxj0yABLOB 16hncO	9995.93 Supplier#000388258 UNITED KINGDOM		
9997.98 Supplier#002053398 RUSSIA	48388257.00 Manufacturer#1		
56803343.00 Manufacturer#5	B7Q0j7Qhy4P4BhOBhMAhxC		33-829-188-2160
lwy156SR47z PiMyi 32-253-632-7713	ijwJx1C3R5NPx01BS4AQ2z2M2QzinSimP25igN6QS17CQmR2710i3NM		
105AgwCzlgnmIAknMnMNYMRCnMkinAi6BlCjgnwA1AkjllLCRzw3	9995.86 Supplier#002202674 ROMANIA		
9997.98 Supplier#002145281 RUSSIA	18702661.00 Manufacturer#2		
56145280.00 Manufacturer#3	6zN5m Bkl2zOC2O		29-465-515-1025
CjLnxjQyONMQhLNQOQ1NL0hOBz6PgzRkhhmR 32-938-997-4594	x31Pza1B5On5wkMwi6BOOnCO06wRR4k5mn30S51NQS7B4M36zgjOC4P4CB hna10j1B2zP24yB430B0YRHOK14wJwA61KXO		
1MQw67hx07OM1RN6CjCQ30Q6AOKmQM0N2	9995.73 Supplier#002166538 ROMANIA		
9997.97 Supplier#000237396 FRANCE	20166537.00 Manufacturer#1		
27237395.00 Manufacturer#3	P5PNh2P5j5m6N0B33		29-412-933-5599
Bgm5zha BnmN0Cz1NMwin0CCRMh 16-450-280-7386	2 0iLm16BL611hON1P67MMmRz N4gn3vniOlh13lOmy24M 41n26MimjLxIN616B2MM42L2nmkjyOZM5AOnxn2mNHL1jB15C		
Ni0M6mgz4CR23CkLjQ5OyPyj1zQaiky5ACmwn3lxii	9995.73 Supplier#002166538 ROMANIA		
9997.82 Supplier#002252880 ROMANIA	24666521.00 Manufacturer#3		
33002846.00 Manufacturer#1	P5PNh2P5j5m6N0B33		29-412-933-5599
w7k4hOLBn4AgyixyN y2xQwvBxQN355PM70g 29-839-604-2322	2 0iLm16BL611hON1P67MMmRz N4gn3vniOlh13lOmy24M 41n26MimjLxIN616B2MM42L2nmkjyOZM5AOnxn2mNHL1jB15C		
306681zgn1068Qim2i5l0hnw5C0AA0wM30B3jigx75SAw3CBlyR2in46A 05BjQ22LSCPg3ncwmm71xniim611	9995.48 Supplier#000188944 GERMANY		
9997.73 Supplier#000329974 RUSSIA	32438933.00 Manufacturer#1		
15329973.00 Manufacturer#3	P7h0zAN4x0z1N1ONQ5C30nNPK		17-127-147-4930
3j1nNhaAR01Si74wx4nLRShlC1Qymj 32-268-488-7178	OQz14zmzSMLy5i05g4hxgw0iBPnyk67zPQwlnh70gRM4k5QA		
mAi2wQ nPijNMORS074AzM7Cim1zyk0k	9995.41 Supplier#002021001 GERMANY		
9997.57 Supplier#001712265 ROMANIA	7270998.00 Manufacturer#3		
3212262.00 Manufacturer#5	7AM P5Pm50gCLS nziSz		17-784-961-9980
PgNSmkzghknSkN0NwyN00lijBykNyzAP35MQ1kz 29-108-652-9788	zAN7O2yQ6L147A0PBmgn5g0xS4g		
xNw24PBOR57kynhn7j2l0j7nk15QMjB0AS0zOwg0PnmLB03w3j	9995.26 Supplier#002022928 ROMANIA		
9997.55 Supplier#000499287 GERMANY	41022927.00 Manufacturer#5		
10999280.00 Manufacturer#5	02xhm5107w13gl6Cm6n7		29-705-698-4233
mCw2RzyM25zhOQP1hg51Pihh Q 17-147-241-4068	gg3kkhAPoiCzn6k52N fz00SB5120ZA6Lw72A6h3mn07h6jnrhng5g1i6L1wNOy k0 3		
MikM2Pk AAMAKmxPkOhvixgkijv4OR 2OmRNwQ NRC320PywBPwB01NzxmMhAjM My0ihhnhQA6PyROMk iLAR0y	9995.25 Supplier#001396488 FRANCE		
9997.06 Supplier#000725417 GERMANY	59896449.00 Manufacturer#4		
11225410.00 Manufacturer#2	27BAQi6mKh5RySQR2lkB50PnQyghL01P		16-529-651-1456
611nllxgA4ig21QA5yCn 17-577-582-2010	10hy3k43ASk521igk6SNmzCjgkMR5MLhwn6hz12kSkmOQngxw12P5mL3CL xA		
6z0j0lkr2Rm6ALnzhrP16hkh6nB6Rm01Mx1xM3g6lx1Q37	9995.21 Supplier#001147030 GERMANY		
9996.87 Supplier#002185033 GERMANY	28897002.00 Manufacturer#2		
7435030.00 Manufacturer#2	l j00BlPwRy4M0h x6C5A2C7Ax5 2S		17-148-758-4770
jkNyzkky P762wj 17-809-292-5756	iyLnnL0ggR1jj42RxnQL400zQ05g1N5P6MyN53xB m50SBN1j7jCn40Bk5BAY6LgX03		
iQ6BA66M N	9995.03 Supplier#001905504 RUSSIA		
n2CRP07LNM1h76PA31j7RkmN5nAkjg0AnN20L4kx03Nzn1Mi5LSzC	44655461.00 Manufacturer#5		
9996.68 Supplier#000209198 RUSSIA	zLRxxMQ3zRC3kQAONljx6N Om6 j		32-291-626-6898
56459179.00 Manufacturer#5	2PySlnnSSAIAAx21xLj1NR5h0zOhhQkzQi4ygzAz0j		
Q7nBSlmQiLylL SkRxsjNg 32-610-484-4619	9994.94 Supplier#001431449 GERMANY		
P70Ah07jLbzRSOQMBynj74kjM yijhhM6n14 x0rImP6P7Lm1h76PA31j7RkmN5nAkjg0AnN20L4kx03Nzn1Mi5LSzC	56931412.00 Manufacturer#2		
M747AL5kByxRjSkS014wn5LiQmAmyLQy615			

LkNPCn1QL 452xiSg2xSwhOh	17-526-346-6654	nBw1M0A4Bh61Allw4S2z2lMP7LxBBO64S2mhs0mm2nQBRLP515A1
gy6B4gzQmnnLLQ6MnSC5OR34RgP3RMzA0Qw5Pm6RCABHSijj5w2zwwQgiLO		9993.84 Supplier#002138572 ROMANIA
9994.82 Supplier#001834220 RUSSIA		4388570.00 Manufacturer#5
14584207.00 Manufacturer#4		k5g 6k6nwCC4CygZPxC6k70PzC1m2iBjB500i 29-284-116-1075
3nP2AyPOPmGQnAn 32-933-374-3196		5AgN3jOgR5j0Bz26MPhhPR6Lnn4m7yQmICAhwlxx107mkz44m1QCwSk2Ck
SNR2MonghO5Q33M52iLOjjiwM62PR23yjmwm		9993.70 Supplier#001508000 RUSSIA
9994.81 Supplier#001961809 GERMANY		7507999.00 Manufacturer#4
28211799.00 Manufacturer#4		OCpX4Lgy7kBW 32-556-927-3607
RjkyiN ih0A5mAP010300x7kgmySn BONMCSB35i 17-251-252-5653		w10L2jBBm315AjlKSCB0hz
LgmX0ONSQm1C54RMN17LwRgSO 60m7L 4w3hLin		BiSjR0m2M2nAsihhRC47xNSrZQO05PjMn6m1zQ6n2zgiSBzN4hQ1k37ALh
9994.80 Supplier#002745721 UNITED KINGDOM		Yn4iN5xnr MGK
495720.00 Manufacturer#2		9993.53 Supplier#000597447 UNITED KINGDOM
50Cg6jxO0zS43n 33-810-127-3228		29847437.00 Manufacturer#1
i2kgBCMAjC ChOyilSLPw7wwAhLC2Cj243SOQ5BLw1On23R1		4i6kmwSijh m2QNNBhQQ 33-986-148-2564
9994.72 Supplier#000810547 GERMANY		xQSSvyywhlnSm02kAZxgLL6mLxh k4L
55560492.00 Manufacturer#3		OR21iAX41QW 0631y332ln055iy4CC2Ck16hRMLhwi
S5P3R57xQgRhQy3lxyzhyzny4y26kzz7S60134 17-323-969-8247		9993.25 Supplier#001030984 FRANCE
Ckyyo1Rwg0NRMnh h0B0NA7wi0nMkr6Lz7y7CSjgkN		37030983.00 Manufacturer#5
9994.67 Supplier#002489390 UNITED KINGDOM		wykw 5mABPQwxhLbJ16gS2Nxlhn li3x6Az5M7SM 16-954-548-8807
31739379.00 Manufacturer#1		QngQPRMPS5MBYQO3y207M4BBP266hLx34AN0M4AgB2SOg1MQjgnjhlz0Mj1
lNm4MkS7253x2nAL5SzbL4OAx 33-706-178-8775		9993.23 Supplier#001320190 ROMANIA
SzB0A0v034g Si3lygwzzjB2C3gh6NxAQAQ6R1LN2nLx45g		58320189.00 Manufacturer#3
25Llnny720Lnyio6		wAQwg4iPln2OSSyRzC24h4y7hSFM 29-356-764-2467
9994.44 Supplier#002524901 FRANCE		zPBm6xwC 7m623x006h475mPwiB47z0nwlOx715Ngh5Rgw
34774889.00 Manufacturer#5		LkR6jGmws01wQlgn4NOH1 8Nkx72NA2
g576CzB3zmSgXkMzNyg 16-311-680-3270		9993.19 Supplier#001678664 UNITED KINGDOM
2xR5L237gwhRM00ygzLAKwywRRINQ7xx7zNy5y15jC6jR7		18178651.00 Manufacturer#3
9994.42 Supplier#000787883 RUSSIA		4glLNI 520iQhPgkxQzBA0i15 li 4A71ik0 33-485-233-1542
57037863.00 Manufacturer#2		70Qn02PAk2zPkSAi7hQzk63NB00746 h7SMjA7LA107LSQ
z5B6yGAbkC0iSkPPiAj 32-164-821-4421		9993.14 Supplier#001035889 GERMANY
6n 7zAv1SgzzM6nh10hzzPMhAx1z0RwhL3z7PQiCBkCyMCz		54285870.00 Manufacturer#4
lBgnZznIwAjznmP2hKNS60SPfNI2Lx		40gzM42h0gmjMly0S3zQ1QRSy6gCLC4mxA0z 17-306-145-6428
9994.35 Supplier#000656604 UNITED KINGDOM		iZC0B4Cxx
36656603.00 Manufacturer#4		hnhMw02z24jA6Mhs
ChRzLzySR3 zzzm7yM6Am0nN Smigw 33-319-512-1645		9993.09 Supplier#000459749 UNITED KINGDOM
LxOSAfk2SS050OR3BinjhChARI0v25kM 506A6k3j02Alx7mg02i		36459748.00 Manufacturer#3
gMyY0R3k3niQzwhMkC73 ySIR 42mLLMwBRh4N		x3nLglLbizxNSx2003jB21P4Q7zRglJ023B6ylh 33-260-433-4996
9994.15 Supplier#000750884 UNITED KINGDOM		w45hC42MS8C01i imp047k0
25500859.00 Manufacturer#1		18041yRhhjBN4j7wz37SLX0Cg41QgP6140x3zxxxi0RA
MR C2xx0L0wznBmh51 yxPA5BnhRCR1 QQ7m0 33-640-820-7791		RROg2hlyhN44h2
Cz0wBMv0		9993.02 Supplier#000496153 UNITED KINGDOM
XALRCCnCB6Sx3BQ4i7z6mR7hL0AzShL4C22Qm4z03CR1nlyk21R7R30		38746140.00 Manufacturer#3
CRIC0IA50BmNBj5NIMBAjNwB6Gh0ZS1		BNy2iS37zQm 33-142-604-4985
9994.11 Supplier#002450182 FRANCE		miNlyQ3OCQRRPh1z5CLOWOS1NBQ6SNSL0iQ1N2725Nk4N7h4B
38450181.00 Manufacturer#1		9993.02 Supplier#001817737 UNITED KINGDOM
67012wMhRxoOolZ Mi42LiPmQz1knlLICOokSwR 16-689-491-7520		49067720.00 Manufacturer#5
NJMMR7LClms		P11Nm1n722Blx7iig OxAMBYNO1Mlmw7jk7lQSQP 33-153-441-6667
7B3zwk74m4iLSOx361sJkQN5C1x7RBOisjy97hg5P71xgik4y1		zC7nj1BPnkhlNiBwimzCkxkj iClxjhlQnAlhL73Bh7ASM
9993.98 Supplier#001355025 GERMANY		9992.98 Supplier#001964661 UNITED KINGDOM
8105018.00 Manufacturer#3		32714630.00 Manufacturer#3
LQ6Aw21SPR NkB hh2mMBML3zML5Rw 17-806-303-1090		4wAQ3j50LMjBgA7nx1QOBRg 4i2 P 33-384-325-2804
RyOALi07gqkiNk72jg2k1AyNPl4Q3iz2hi4Nck6Cw55C6Mgi110yJL6w3xN		MPHhx5xZAmu3Sv1
9993.98 Supplier#001355025 GERMANY		hNlHgQmk11
47104979.00 Manufacturer#2		9992.92 Supplier#000262473 GERMANY
LQ6Aw21SPR NkB hh2mMBML3zML5Rw 17-806-303-1090		8512470.00 Manufacturer#5
RyOALi07gqkiNk72jg2k1AyNPl4Q3iz2hi4Nck6Cw55C6Mgi110yJL6w3xN		k14jzL547M7Mhh 17-652-782-6436
9993.98 Supplier#000831504 ROMANIA		MSvno05CjPMLLL3610i0651z64An51MvC03C
5331501.00 Manufacturer#1		5BLlJbyi07Nhp54ginsCz005knbIC332Nn6A
yCB7mAmNBPLkxBjwzxiQgAnSC 29-775-353-7830		9992.86 Supplier#001956454 RUSSIA
2RPgAl56iOQL3hBwRQ0zSO5nCSQx42jMz3wi0m02iB1nlPwkQng		29706426.00 Manufacturer#5
9993.98 Supplier#000831504 ROMANIA		S BmB7z7wwSR4SLnRQ1640C 32-501-576-4753
51831503.00 Manufacturer#1		4m2Anm3Ng10CS106zmRBCh6xhQRhCSPLwg2B
yCB7mAmNBPLkxBjwzxiQgAnSC 29-775-353-7830		9992.83 Supplier#002794535 GERMANY
2RPgAl56iOQL3hBwRQ0zSO5nCSQx42jMz3wi0m02iB1nlPwkQng		25294518.00 Manufacturer#1
9993.97 Supplier#000566725 FRANCE		zkjJkny 7674CL3xLMLw1CAOXwzOCOMR 17-974-853-7096
15566724.00 Manufacturer#1		34516AyR5jA3S3SMkAk 24yB61P3kms5nmzB0yRn
ChRmKR mS6zN 16-151-920-7745		40zCL3gn0AkgozgzLwh2761gi7hJp3x
i3xw30LiPA37j1lw7mNBBLOR7k4517BLL42mRQILN2n		9992.71 Supplier#000401090 GERMANY
g1gnC1Bg02h6j0410CA01M 0z51zQjNAw1gComxMg04jzxlgAnPi4L		31901069.00 Manufacturer#1
9993.91 Supplier#000674336 RUSSIA		ylyxLz Qym23Lym3 17-358-630-1671
14174327.00 Manufacturer#4		k6CwhOjkk3BBM6yRPyAmxA0117BCAAGslnj4kP7ny7Nj00L60y1L3AkxA
3 57R6Min lz4i64N4S1QLP0B0BnMnjxhzi 32-949-587-4877		9992.70 Supplier#000245718 UNITED KINGDOM
01RR1RR3Nh3Q02ByBxhx		995717.00 Manufacturer#5
Z713Cy3R7ML61Cxx14z2NzxLziyOnln50M0zwsjM2QSnx0zs 5AR2CQ		N4 1lk6yw5 Og07M4Rn02ihBm7z7lhygx 33-570-729-4176
9993.90 Supplier#000692154 UNITED KINGDOM		0 hb
19442135.00 Manufacturer#1		3SR3SLN6ywl1Lln4AQxJAC1m84
044j0khp760Ax0k03633NPM1SAhP7LyOwRyLjih 33-849-416-5457		9992.66 Supplier#000592271 GERMANY
LR27yC2 4RnLB41klnRhOwkxjR2QnxlRj		49342222.00 Manufacturer#1
9993.85 Supplier#0002720604 FRANCE		CMj6w PNOc jm32zgn3mQhMQiMimgj5jBk0zLN5m 17-360-302-3279
16220593.00 Manufacturer#3		nhy0OnxNC12xzMgBOAijSP607AMnQyH0kzOPgLiRCw1h
MOLSK5ABNS1L5i2Lg761xinlL7 16-970-958-1127		9992.66 Supplier#002666275 UNITED KINGDOM
nBw1M0A4Bh61Allw4S2z2lMP7LxBBO64S2mhs0mm2nQBRLP515A1		37916262.00 Manufacturer#3
9993.85 Supplier#002720604 FRANCE		2kihi7LRkizAQ3P 33-232-137-3532
48470555.00 Manufacturer#4		MCLiN34SPiMxgR 1614AiMPLA Mk2BzhjRw1gn
MOLSK5ABNS1L5i2Lg761xinlL7 16-970-958-1127		hS6S2gPA416PLB07Cg1xzwni1jhx5h5S
		9992.62 Supplier#000483146 UNITED KINGDOM

```

23733138.00      Manufacturer#4      Brand#41      SMALL BURNISHED NICKEL      49.00
wxkLnPP0miimg0xOhSizlAm3 jB2      33-955-934-3870      Brand#112     SMALL PLATED TIN      49.00
x5YkxL7x nLLPyn LP0m xLPnLmjz5Cn3R6CAy70y1hASwlmC7mQ1PB      Brand#52     PROMO BURNISHED STEEL      48.00
xw5h45g9Q6L7      Supplier#000099650      RUSSIA      Brand#54     PROMO BRUSHED STEEL      37.00
6849643.00      Manufacturer#3      Brand#13     LARGE ANODIZED BRASS      49.00
M3jM0kP5izxS2zMml06y31Ly70kn 76iyBm      32-971-481-2533      Brand#21     LARGE POLISHED COPPER      47.00
vAP0mONL2nwy mOzz4ziczwR 64xj2M5Aw0      Brand#35     SMALL BURNISHED STEEL      31.00
MOM1Z1IMPZ0LCIRMX6jNMNBLMhz1      Brand#21     MEDIUM PLATED COPPER      20.00
9992.54      Supplier#000099650      RUSSIA      Brand#12     LARGE ANODIZED COPPER      45.00
37599625.00      Manufacturer#4      Brand#35     PROMO BRUSHED BRASS      49.00
M3jM0kP5izxS2zMml06y31Ly70kn 76iyBm      32-971-481-2533      Brand#42     SMALL BRUSHED TIN      45.00
vAP0mONL2nwy mOzz4ziczwR 64xj2M5Aw0      Brand#34     MEDIUM PLATED NICKEL      48.00
MOM1Z1IMPZ0LCIRMX6jNMNBLMhz1      Brand#34     PROMO ANODIZED TIN      45.00
9992.41      Supplier#000853625      UNITED KINGDOM      Brand#35     PROMO ANODIZED COPPER      49.00
11353618.00      Manufacturer#2      Brand#22     STANDARD PLATED NICKEL      20.00
Akyx 1Nk2Mijgn6CAABAlw5660      33-590-481-6043      Brand#54     SMALL ANODIZED NICKEL      45.00
PjgjCk7gR21wB1ABow5ySzA5R4B0R1MChljPw0gL4ymCOigwzGR      Brand#21     PROMO BRUSHED TIN      31.00
9992.40      Supplier#002540732      RUSSIA      Brand#25     ECONOMY BURNISHED NICKEL      31.00
31790721.00      Manufacturer#1      Brand#12     LARGE BRUSHED COPPER      45.00
MP2hB5w3m05xL      32-855-640-9841      Brand#24     SMALL BRUSHED STEEL      48.00
mk624hpc136M20i3Q14yzzCPPNgClSPai7BRQOOM3wz0zmN2wS6zlhNxCB      Brand#35     STANDARD BRUSHED BRASS      48.00
LLZZSA664M1S6      Brand#43     PROMO ANODIZED COPPER      37.00

```

100 rows processed.

Statement Processed in 414.40 seconds.

Ended Executing this Query at Fri Mar 7 23:30:26 1997

Query Started at 857805811.73

Query Ended at 857806226.13

Query Processed in 414.40 seconds.

-- Query 16 (Standard)

Began Executing this Query at Fri Mar 7 23:30:26 1997

```

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#14'
AND P_TYPE NOT LIKE 'ECONOMY PLATED%'
AND P_SIZE IN (47, 31 ,49 ,48 ,37, 45, 20, 11)
AND PS_SUPPKEY NOT IN (SELECT S_SUPPKEY
FROM SUPPLIER
WHERE S_COMMENT LIKE '%Better Business
Bureau%complaints%')
GROUP BY P_BRAND, P_TYPE, P_SIZE
ORDER BY SUPPLIER_CNT DESC, P_BRAND, P_TYPE, P_SIZE

```

P_BRAND	P_TYPE	P_SIZE
Brand#12	SMALL POLISHED COPPER	45.00
Brand#32	LARGE BURNISHED TIN	49.00
Brand#11	PROMO POLISHED NICKEL	11.00
Brand#45	LARGE POLISHED STEEL	31.00
Brand#51	MEDIUM BRUSHED COPPER	11.00
Brand#23	SMALL ANODIZED NICKEL	49.00
Brand#11	MEDIUM POLISHED STEEL	20.00
Brand#51	ECONOMY ANODIZED NICKEL	31.00
Brand#43	ECONOMY BRUSHED COPPER	48.00
Brand#51	ECONOMY ANODIZED TIN	47.00
Brand#52	MEDIUM ANODIZED TIN	31.00
Brand#31	LARGE POLISHED COPPER	37.00
Brand#32	STANDARD PLATED STEEL	20.00
Brand#22	PROMO BRUSHED TIN	49.00
Brand#53	LARGE BURNISHED TIN	31.00
Brand#35	LARGE PLATED BRASS	48.00
Brand#41	SMALL BURNISHED NICKEL	49.00
Brand#12	SMALL PLATED TIN	49.00
Brand#52	PROMO BURNISHED STEEL	48.00
Brand#54	PROMO BRUSHED STEEL	37.00
Brand#13	LARGE ANODIZED BRASS	49.00
Brand#21	LARGE POLISHED COPPER	47.00
Brand#35	SMALL BURNISHED STEEL	31.00
Brand#21	MEDIUM PLATED COPPER	20.00
Brand#12	LARGE ANODIZED COPPER	45.00
Brand#35	PROMO BRUSHED BRASS	49.00
Brand#42	SMALL BRUSHED TIN	45.00
Brand#34	MEDIUM PLATED NICKEL	48.00
Brand#34	PROMO ANODIZED TIN	45.00
Brand#35	PROMO ANODIZED COPPER	49.00
Brand#22	STANDARD PLATED NICKEL	20.00
Brand#54	SMALL ANODIZED NICKEL	45.00
Brand#21	PROMO BRUSHED TIN	31.00
Brand#25	ECONOMY BURNISHED NICKEL	31.00
Brand#12	LARGE BRUSHED COPPER	45.00
Brand#24	SMALL BRUSHED STEEL	48.00
Brand#35	STANDARD BRUSHED BRASS	48.00
Brand#43	PROMO ANODIZED COPPER	37.00
Brand#53	STANDARD BURNISHED TIN	45.00
Brand#23	LARGE PLATED NICKEL	20.00
Brand#25	MEDIUM ANODIZED COPPER	45.00
Brand#34	LARGE PLATED TIN	11.00
Brand#42	ECONOMY POLISHED COPPER	31.00
Brand#52	SMALL PLATED TIN	11.00
Brand#41	PROMO BRUSHED COPPER	45.00
Brand#11	ECONOMY POLISHED TIN	48.00
Brand#15	MEDIUM PLATED NICKEL	47.00
Brand#24	STANDARD BRUSHED NICKEL	45.00
Brand#52	MEDIUM BURNISHED TIN	49.00
Brand#52	STANDARD BRUSHED NICKEL	45.00
Brand#11	ECONOMY BURNISHED COPPER	31.00
Brand#11	LARGE POLISHED STEEL	37.00
Brand#24	PROMO BURNISHED STEEL	11.00
Brand#32	ECONOMY ANODIZED STEEL	45.00
Brand#12	LARGE PLATED TIN	48.00
Brand#41	MEDIUM ANODIZED TIN	47.00
Brand#13	PROMO BRUSHED BRASS	45.00
Brand#32	STANDARD PLATED NICKEL	20.00
Brand#35	PROMO BRUSHED COPPER	47.00
Brand#41	SMALL BRUSHED BRASS	49.00
Brand#54	ECONOMY BURNISHED NICKEL	37.00
Brand#51	LARGE BURNISHED TIN	45.00
Brand#11	ECONOMY ANODIZED COPPER	11.00
Brand#15	LARGE PLATED TIN	11.00
Brand#21	STANDARD POLISHED BRASS	20.00
Brand#25	ECONOMY POLISHED STEEL	47.00
Brand#35	PROMO BURNISHED STEEL	37.00
Brand#35	STANDARD POLISHED TIN	48.00
Brand#44	MEDIUM PLATED COPPER	48.00
Brand#52	LARGE PLATED STEEL	37.00
Brand#55	PROMO POLISHED BRASS	47.00

Brand#42 1479.00	SMALL ANODIZED TIN	49.00	Brand#41 1466.00	LARGE BURNISHED STEEL	45.00
Brand#54 1479.00	STANDARD BURNISHED STEEL	45.00	Brand#52 1466.00	MEDIUM PLATED NICKEL	20.00
Brand#55 1479.00	SMALL BRUSHED COPPER	48.00	Brand#13 1465.00	SMALL BRUSHED NICKEL	37.00
Brand#31 1478.00	LARGE BURNISHED STEEL	31.00	Brand#12 1464.00	MEDIUM BURNISHED TIN	11.00
Brand#31 1477.00	STANDARD PLATED BRASS	11.00	Brand#15 1464.00	STANDARD BURNISHED COPPER	20.00
Brand#34 1477.00	PROMO BRUSHED COPPER	20.00	Brand#23 1464.00	STANDARD ANODIZED TIN	31.00
Brand#22 1476.00	MEDIUM ANODIZED TIN	11.00	Brand#24 1464.00	STANDARD BRUSHED BRASS	37.00
Brand#24 1476.00	SMALL BURNISHED NICKEL	45.00	Brand#32 1464.00	MEDIUM POLISHED BRASS	48.00
Brand#25 1476.00	STANDARD POLISHED BRASS	49.00	Brand#42 1464.00	PROMO PLATED NICKEL	31.00
Brand#34 1476.00	SMALL ANODIZED COPPER	11.00	Brand#52 1464.00	LARGE BRUSHED TIN	20.00
Brand#41 1476.00	SMALL BURNISHED NICKEL	11.00	Brand#11 1463.00	LARGE BURNISHED NICKEL	47.00
Brand#53 1476.00	ECONOMY ANODIZED COPPER	45.00	Brand#11 1463.00	PROMO BURNISHED STEEL	20.00
Brand#54 1476.00	LARGE BRUSHED COPPER	49.00	Brand#25 1463.00	LARGE BURNISHED COPPER	20.00
Brand#54 1476.00	SMALL ANODIZED NICKEL	20.00	Brand#32 1463.00	LARGE ANODIZED COPPER	31.00
Brand#13 1475.00	STANDARD BRUSHED BRASS	45.00	Brand#33 1463.00	LARGE ANODIZED STEEL	45.00
Brand#15 1475.00	SMALL ANODIZED COPPER	47.00	Brand#34 1463.00	MEDIUM ANODIZED BRASS	48.00
Brand#24 1475.00	PROMO ANODIZED TIN	11.00	Brand#55 1463.00	LARGE BRUSHED COPPER	47.00
Brand#25 1475.00	LARGE ANODIZED COPPER	11.00	Brand#22 1462.00	ECONOMY ANODIZED COPPER	31.00
Brand#22 1474.00	PROMO BRUSHED NICKEL	20.00	Brand#24 1462.00	STANDARD BURNISHED BRASS	47.00
Brand#22 1474.00	SMALL BURNISHED TIN	11.00	Brand#25 1462.00	ECONOMY BRUSHED STEEL	31.00
Brand#23 1474.00	LARGE ANODIZED COPPER	31.00	Brand#31 1462.00	ECONOMY BURNISHED COPPER	47.00
Brand#52 1474.00	STANDARD ANODIZED TIN	48.00	Brand#32 1462.00	MEDIUM POLISHED COPPER	11.00
Brand#21 1473.00	MEDIUM BRUSHED TIN	37.00	Brand#35 1462.00	SMALL BRUSHED NICKEL	31.00
Brand#35 1473.00	STANDARD ANODIZED COPPER	31.00	Brand#44 1462.00	SMALL ANODIZED BRASS	49.00
Brand#44 1473.00	SMALL BURNISHED COPPER	37.00	Brand#54 1461.00	PROMO PLATED STEEL	11.00
Brand#15 1472.00	STANDARD BURNISHED BRASS	49.00	Brand#13 1460.00	STANDARD ANODIZED NICKEL	48.00
Brand#22 1472.00	MEDIUM BURNISHED NICKEL	11.00	Brand#25 1460.00	STANDARD ANODIZED COPPER	47.00
Brand#23 1472.00	LARGE ANODIZED BRASS	37.00	Brand#31 1460.00	MEDIUM BURNISHED COPPER	47.00
Brand#32 1472.00	LARGE BRUSHED BRASS	31.00	Brand#33 1460.00	ECONOMY BURNISHED COPPER	48.00
Brand#42 1472.00	PROMO BURNISHED NICKEL	47.00	Brand#41 1460.00	LARGE POLISHED STEEL	31.00
Brand#51 1472.00	PROMO POLISHED STEEL	37.00	Brand#52 1460.00	STANDARD BRUSHED COPPER	31.00
Brand#15 1471.00	STANDARD BURNISHED STEEL	45.00	Brand#54 1460.00	PROMO ANODIZED NICKEL	45.00
Brand#21 1471.00	STANDARD BURNISHED TIN	48.00	Brand#15 1459.00	STANDARD POLISHED STEEL	47.00
Brand#22 1471.00	ECONOMY POLISHED COPPER	47.00	Brand#25 1459.00	SMALL ANODIZED TIN	20.00
Brand#22 1471.00	LARGE ANODIZED TIN	37.00	Brand#25 1459.00	STANDARD PLATED STEEL	11.00
Brand#31 1471.00	ECONOMY BURNISHED COPPER	45.00	Brand#32 1459.00	SMALL ANODIZED BRASS	31.00
Brand#44 1471.00	SMALL POLISHED STEEL	47.00	Brand#34 1459.00	PROMO PLATED BRASS	37.00
Brand#54 1471.00	ECONOMY ANODIZED TIN	37.00	Brand#43 1459.00	SMALL PLATED BRASS	31.00
Brand#34 1470.00	PROMO POLISHED COPPER	11.00	Brand#55 1459.00	MEDIUM PLATED NICKEL	11.00
Brand#41 1469.00	MEDIUM PLATED BRASS	48.00	Brand#23 1458.00	ECONOMY BRUSHED BRASS	48.00
Brand#13 1468.00	SMALL BRUSHED STEEL	20.00	Brand#33 1458.00	STANDARD BURNISHED COPPER	48.00
Brand#15 1468.00	ECONOMY BURNISHED NICKEL	48.00	Brand#52 1458.00	SMALL BURNISHED COPPER	20.00
Brand#24 1468.00	MEDIUM PLATED COPPER	20.00	Brand#55 1458.00	ECONOMY BURNISHED NICKEL	31.00
Brand#31 1468.00	STANDARD POLISHED STEEL	37.00	Brand#24 1457.00	LARGE BURNISHED NICKEL	49.00
Brand#32 1468.00	LARGE BRUSHED TIN	49.00	Brand#11 1456.00	SMALL ANODIZED COPPER	37.00
Brand#35 1468.00	STANDARD PLATED BRASS	48.00	Brand#12 1456.00	PROMO BURNISHED BRASS	45.00
Brand#44 1468.00	SMALL PLATED STEEL	20.00	Brand#15 1456.00	ECONOMY BRUSHED BRASS	48.00
Brand#45 1468.00	LARGE ANODIZED NICKEL	49.00	Brand#21 1456.00	LARGE BURNISHED STEEL	31.00
Brand#54 1468.00	PROMO PLATED STEEL	45.00	Brand#22 1456.00	MEDIUM POLISHED BRASS	31.00
Brand#55 1468.00	PROMO PLATED BRASS	20.00	Brand#31 1456.00	PROMO BURNISHED BRASS	48.00
Brand#15 1467.00	MEDIUM PLATED TIN	45.00	Brand#33 1456.00	PROMO PLATED NICKEL	47.00
Brand#24 1467.00	ECONOMY POLISHED COPPER	45.00	Brand#35 1456.00	SMALL POLISHED COPPER	11.00
Brand#24 1467.00	PROMO ANODIZED BRASS	20.00	Brand#45 1456.00	SMALL BURNISHED BRASS	47.00
Brand#54 1467.00	SMALL BURNISHED NICKEL	48.00	Brand#51 1456.00	ECONOMY POLISHED BRASS	31.00
Brand#32 1466.00	PROMO BURNISHED NICKEL	49.00	Brand#12 1455.00	ECONOMY BURNISHED COPPER	20.00

```

Brand#22 ECONOMY BURNISHED BRASS 48.00 1 row processed.
1455.00 Statement Processed in 0.01 seconds.
Brand#25 LARGE ANODIZED BRASS 37.00
1455.00 DROP TABLE ALL_SALES0
Brand#25 LARGE POLISHED STEEL 37.00 Statement Processed in 0.33 seconds.
Brand#31 SMALL ANODIZED STEEL 11.00
1455.00 DROP TABLE SUM_PROMO_SALES0
Brand#32 PROMO POLISHED STEEL 20.00 Statement Processed in 0.14 seconds.
Brand#32 SMALL POLISHED BRASS 37.00
1455.00 DROP TABLE SUM_ALL_SALES0
Brand#33 ECONOMY BURNISHED BRASS 20.00 Statement Processed in 0.15 seconds.
1455.00
Brand#33 STANDARD ANODIZED STEEL 45.00
1455.00 Ended Executing this Query at Fri Mar 7 23:42:02 1997
Brand#34 ECONOMY BRUSHED NICKEL 37.00
1455.00
Brand#35 PROMO ANODIZED BRASS 37.00
1455.00
Brand#35 PROMO POLISHED STEEL 31.00 Query Started at 857806523.12
1455.00 Query Ended at 857806922.10
Brand#41 SMALL BURNISHED NICKEL 47.00 Query Processed in 398.99 seconds.
1455.00
Brand#42 MEDIUM PLATED NICKEL 47.00
1455.00
Brand#42 SMALL BURNISHED TIN 31.00
1455.00
Brand#44 STANDARD ANODIZED BRASS 47.00
1455.00
Brand#45 LARGE BRUSHED BRASS 49.00
1455.00
Brand#45 MEDIUM BRUSHED BRASS 37.00
1455.00
Brand#51 SMALL PLATED STEEL 37.00
1455.00
Brand#23 ECONOMY BRUSHED NICKEL 48.00
1454.00
27840 rows processed.
Statement Processed in 296.98 seconds.
Ended Executing this Query at Fri Mar 7 23:35:23 1997

Query Started at 857806226.13
Query Ended at 857806523.12
Query Processed in 296.98 seconds.

-----
-- Query 8 (Variant B)

Began Executing this Query at Fri Mar 7 23:42:02 1997

SELECT
YEAR,
SUM(DECODE(NATION, 'CANADA', VOLUME, 0)) / SUM(VOLUME) AS
MKT_SHARE
FROM
(SELECT
TO_CHAR(O_ORDERDATE,'YYYY') AS YEAR,
L_EXTENDEDPRISE*(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 PLATED TIN'
) ALL_NATIONS
GROUP BY YEAR
ORDER BY YEAR

YEAR
MKT_SHARE
1995
0.04
1996
0.04

2 rows processed.
Statement Processed in 324.14 seconds.
Ended Executing this Query at Fri Mar 7 23:47:26 1997

Query Started at 857806922.10
Query Ended at 857807246.24
Query Processed in 324.14 seconds.

-----
-- Query 12 (Variant B)

Began Executing this Query at Fri Mar 7 23:47:26 1997

SELECT
L_SHIPMODE,
SUM(DECODE(O_ORDERPRIORITY,'1-URGENT',1,'2-HIGH',1,0)) AS
HIGH_LINE_COUNT,
PROMO_REVENUE
16.67

```

```

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 ('AIR','FOB')
AND L_COMMITDATE < L_RECEIPTDATE
AND L_SHIPDATE < L_COMMITDATE
AND L_RECEIPTDATE >= TO_DATE('1996-01-01','YYYY-MM-DD')
AND L_RECEIPTDATE < ADD_MONTHS(TO_DATE('1996-01-01',
'YYYY-MM-DD'),12)
GROUP BY L_SHIPMODE
ORDER BY L_SHIPMODE

```

L_SHIPMODE	HIGH_LINE_COUNT	LOW_LINE_COUNT
AIR	1875235.00	2816639.00
FOB	1876581.00	2817891.00

2 rows processed.

Statement Processed in 484.69 seconds.

Ended Executing this Query at Fri Mar 7 23:55:30 1997

Query Started at 857807246.24

Query Ended at 857807730.93

Query Processed in 484.69 seconds.

-- Query 17 (Variant B)

Began Executing this Query at Fri Mar 7 23:55:30 1997

```

SELECT SUM(L_EXTENDEDPRI) / 7.0 AS AVG_YEARLY
FROM LINEITEM, (SELECT
P_PARTKEY, 0.2 * AVG(L_QUANTITY) AS AVGQTY
FROM PARTS, LINEITEM
WHERE P_PARTKEY = L_PARTKEY
AND P_BRAND = 'Brand#14'
AND P_CONTAINER = 'JUMBO CAN'
GROUP BY P_PARTKEY) AVG_QUANTITY
WHERE
P_PARTKEY = L_PARTKEY
AND L_QUANTITY < AVGQTY

```

AVG_YEARLY
99838451.34

1 row processed.

Statement Processed in 71.82 seconds.

Setting the number of rows to fetch to: 10

Ended Executing this Query at Fri Mar 7 23:56:42 1997

Query Started at 857807730.93

Query Ended at 857807802.76

Query Processed in 71.83 seconds.

-- Query 3 (Standard)

Began Executing this Query at Fri Mar 7 23:56:42 1997

```

SELECT
L_ORDERKEY,
SUM(L_EXTENDEDPRI*(1-L_DISCOUNT)) AS REVENUE,
TO_CHAR(O_ORDERDATE, 'YYYY-MM-DD'),
O_SHIPRIORITY
FROM CUSTOMER, ORDERS, LINEITEM
WHERE C_MKTSEGMENT = 'AUTOMOBILE'
AND C_CUSTKEY = O_CUSTKEY
AND L_ORDERKEY = O_ORDERKEY
AND O_ORDERDATE < TO_DATE('1995-03-23','YYYY-MM-DD')
AND L_SHIPDATE > TO_DATE('1995-03-23','YYYY-MM-DD')
GROUP BY L_ORDERKEY, O_ORDERDATE, O_SHIPRIORITY

```

ORDER BY REVENUE DESC, O_ORDERDATE

L_ORDERKEY	REVENUE
TO_CHAR(O_ORDER O_SHIPRIORITY 1444780707.00	512680.28
1995-03-02	0.00
3082914.00	500658.85
1995-03-07	0.00
101866116.00	494315.10
1995-03-20	0.00
1441951040.00	493458.23
1995-03-19	0.00
939065283.00	487270.43
1995-03-18	0.00
1559827907.00	486977.14
1995-02-26	0.00
1593220485.00	482665.30
1995-03-15	0.00
34893956.00	481691.07
1995-03-20	0.00
552923042.00	480046.17
1995-03-12	0.00
1654486215.00	478055.66
1995-03-17	0.00

10 rows processed.

Statement Processed in 769.07 seconds.

Ended Executing this Query at Sat Mar 8 00:09:31 1997

Query Started at 857807802.76

Query Ended at 857808571.83

Query Processed in 769.07 seconds.

-- Query 5 (Standard)

Began Executing this Query at Sat Mar 8 00:09:31 1997

```

SELECT
N_NAME,
SUM(L_EXTENDEDPRI*(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 = 'AFRICA'
AND O_ORDERDATE >= TO_DATE('1996-01-01','YYYY-MM-DD')
AND O_ORDERDATE < ADD_MONTHS(TO_DATE('1996-01-01','YYYY-MM-  
DD'),±2)
GROUP BY N_NAME
ORDER BY REVENUE DESC

```

N_NAME	REVENUE
MOROCCO	15996804362.70
MOZAMBIQUE	15945725501.84
ETHIOPIA	15935804928.45
ALGERIA	15912359683.80
KENYA	15806310902.41

5 rows processed.

Statement Processed in 1404.21 seconds.

Ended Executing this Query at Sat Mar 8 00:32:56 1997

Query Started at 857808571.83
Query Ended at 857809976.04
Query Processed in 1404.21 seconds.

-- Query 13 (Standard)

Began Executing this Query at Sat Mar 8 00:32:56 1997

```
SELECT
YEAR,
SUM(REVENUE) AS REVENUE
FROM
(SELECT
TO_CHAR(O_ORDERDATE,'YYYY') AS YEAR,
L_EXTENDEDPRI * (1-L_DISCOUNT) AS REVENUE
FROM LINEITEM, ORDERS
WHERE O_ORDERKEY = L_ORDERKEY
AND O_CLERK = 'Clerk#000000156'
AND L_RETURNFLAG = 'R'
) PERFORMANCE
GROUP BY YEAR
ORDER BY YEAR
```

YEAR
REVENUE
1992
15221990.59
1993
16532890.99
1994
17992544.25
1995
4572282.16

4 rows processed.
Statement Processed in 40.68 seconds.

Ended Executing this Query at Sat Mar 8 00:33:36 1997

Query Started at 857809976.04
Query Ended at 857810016.72
Query Processed in 40.68 seconds.

-- Query 7 (Standard)

Began Executing this Query at Sat Mar 8 00:33:36 1997

```
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_EXTENDEDPRI * (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 = 'CANADA' AND N2.N_NAME = 'CHINA') OR
(N1.N_NAME = 'CHINA' AND N2.N_NAME = 'CANADA'))
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
YEAR
REVENUE
CANADA
1995
15910561575.49
CANADA
1996
15987605463.84
CHINA
1995
15873512189.96
CHINA
1996
15940472888.75

4 rows processed.
Statement Processed in 890.57 seconds.

Ended Executing this Query at Sat Mar 8 00:48:27 1997

Query Started at 857810016.72
Query Ended at 857810907.29
Query Processed in 890.58 seconds.

-- Query 9 (Standard)

Began Executing this Query at Sat Mar 8 00:48:27 1997

```
SELECT
NATION,
YEAR,
SUM(AMOUNT) AS SUM_PROFIT
FROM
(SELECT
N_NAME AS NATION,
TO_CHAR(O_ORDERDATE,'YYYY') AS YEAR,
L_EXTENDEDPRI * (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 '%chiffon%'
) PROFIT
GROUP BY NATION, YEAR
ORDER BY NATION, YEAR DESC
```

NATION
YEAR
SUM_PROFIT
ALGERIA
1998
7973957978.02
ALGERIA
1997
13636059345.01
ALGERIA
1996
13667833482.35
ALGERIA
1995
13645517761.79
ALGERIA
1994
13688096076.53
ALGERIA
1993
13671299111.64
ALGERIA
1992
13686912186.80
ARGENTINA

1998	EGYPT
8014516455.68	1998
ARGENTINA	8040047858.28
1997	EGYPT
13676171690.42	1997
ARGENTINA	13729668355.51
1996	EGYPT
13701419866.17	1996
ARGENTINA	13798926398.53
1995	EGYPT
13679206848.63	1995
ARGENTINA	13714206536.09
1994	EGYPT
13675017413.60	1994
ARGENTINA	13743362304.88
1993	EGYPT
13681821878.85	1993
ARGENTINA	13753383624.49
1992	EGYPT
13660401291.23	1992
BRAZIL	13779460885.16
1998	ETHIOPIA
8084108139.61	1998
BRAZIL	8068347638.95
1997	ETHIOPIA
13749938290.27	1997
BRAZIL	13732396906.40
1996	ETHIOPIA
13805847075.30	1996
BRAZIL	13761926283.66
1995	ETHIOPIA
13739722866.41	1995
BRAZIL	13700513527.45
1994	ETHIOPIA
13735883052.86	1994
BRAZIL	13716764991.89
1993	ETHIOPIA
13709372240.45	1993
BRAZIL	13725856716.35
1992	ETHIOPIA
13757920762.08	1992
CANADA	13746708739.55
1998	FRANCE
8044151398.09	1998
CANADA	8062845173.31
1997	FRANCE
13727724132.16	1997
CANADA	13771753107.44
1996	FRANCE
13738527810.28	1996
CANADA	13794399265.00
1995	FRANCE
13751106127.04	1995
CANADA	13738337102.76
1994	FRANCE
13742968141.17	1994
CANADA	13741031421.79
1993	FRANCE
13730226493.48	1993
CANADA	13799696569.87
1992	FRANCE
13745953288.90	1992
CHINA	13818164261.66
1998	GERMANY
8044444994.83	1998
CHINA	8058655841.55
1997	GERMANY
13704104289.60	1997
CHINA	13732360596.93
1996	GERMANY
13683553252.16	1996
CHINA	13789258815.16
1995	GERMANY
13680550257.00	1995
CHINA	13758441321.78
1994	GERMANY
13753992694.80	1994
CHINA	13752156967.04
1993	GERMANY
13684403655.96	1993
CHINA	13776764347.07
1992	GERMANY
13721109098.17	1992

13791175650.04	1992
INDIA	13720847991.71
1998	JAPAN
8069059943.55	1998
INDIA	8008912467.87
1997	JAPAN
13679628865.76	1997
INDIA	13644513594.78
1996	JAPAN
13740991642.39	1996
INDIA	13629290255.48
1995	JAPAN
13746243192.48	1995
INDIA	13642441984.09
1994	JAPAN
13739592144.19	1994
INDIA	13629715137.52
1993	JAPAN
13748244008.65	1993
INDIA	13607079898.70
1992	JAPAN
13770069962.72	1992
INDONESIA	13690260660.17
1998	JORDAN
8029533938.42	1998
INDONESIA	7970983476.83
1997	JORDAN
13695964219.91	1997
INDONESIA	13624583655.06
1996	JORDAN
13734516942.54	1996
INDONESIA	13647879224.11
1995	JORDAN
13668136968.53	1995
INDONESIA	13624885341.80
1994	JORDAN
13666113952.65	1994
INDONESIA	13637290952.99
1993	JORDAN
13663566561.69	1993
INDONESIA	13606626947.84
1992	JORDAN
13700496983.47	1992
IRAN	13643277462.03
1998	KENYA
8046676876.98	1998
IRAN	8031042736.91
1997	KENYA
13717430067.94	1997
IRAN	13676094575.83
1996	KENYA
13758697420.24	1996
IRAN	13739922019.92
1995	KENYA
13702644157.41	1995
IRAN	13653105583.59
1994	KENYA
13715677534.45	1994
IRAN	13640516494.63
1993	KENYA
13680820546.45	1993
IRAN	13693101506.50
1992	KENYA
13679047622.05	1992
IRAQ	13722704493.78
1998	MOROCCO
8027483700.09	1998
IRAQ	8071432102.30
1997	MOROCCO
13704803978.90	1997
IRAQ	13786157332.87
1996	MOROCCO
13760753639.10	1996
IRAQ	13744343915.96
1995	MOROCCO
13680349533.63	1995
IRAQ	13801258022.06
1994	MOROCCO
13743877843.47	1994
IRAQ	13721957201.62
1993	MOROCCO
13725600967.84	1993
IRAQ	13702615484.40

MOROCCO	13701707102.16
1992	RUSSIA
13765573700.48	1992
MOZAMBIQUE	13780280196.58
1998	SAUDI ARABIA
8039104836.95	1998
MOZAMBIQUE	8049732870.41
1997	SAUDI ARABIA
13700667164.20	1997
MOZAMBIQUE	13724262810.21
1996	SAUDI ARABIA
13770647749.76	1996
MOZAMBIQUE	13742160415.74
1995	SAUDI ARABIA
13696594220.58	1995
MOZAMBIQUE	13770629524.43
1994	SAUDI ARABIA
13690062308.91	1994
MOZAMBIQUE	13732793358.91
1993	SAUDI ARABIA
13666958747.49	1993
MOZAMBIQUE	13767608967.89
1992	SAUDI ARABIA
13743978502.91	1992
PERU	13832688241.33
1998	UNITED KINGDOM
8046989438.43	1998
PERU	8115480105.56
1997	UNITED KINGDOM
13721714238.56	1997
PERU	13808760902.18
1996	UNITED KINGDOM
13773539995.89	1996
PERU	13838810367.51
1995	UNITED KINGDOM
13713476050.63	1995
PERU	13834077462.20
1994	UNITED KINGDOM
13725975156.52	1994
PERU	13837659044.62
1993	UNITED KINGDOM
13700350246.53	1993
PERU	13832915088.91
1992	UNITED KINGDOM
13739088260.59	1992
ROMANIA	13862049679.22
1998	UNITED STATES
7992034479.51	1998
ROMANIA	8006504039.36
1997	UNITED STATES
13709582634.98	1997
ROMANIA	13642017478.72
1996	UNITED STATES
13716830155.55	1996
ROMANIA	13671941717.81
1995	UNITED STATES
13639321523.84	1995
ROMANIA	13687721858.15
1994	UNITED STATES
13687172651.19	1994
ROMANIA	13663876308.89
1993	UNITED STATES
13642772270.64	1993
ROMANIA	13648608105.61
1992	UNITED STATES
13724152046.35	1992
RUSSIA	13692644398.52
1998	VIETNAM
8026169576.24	1998
RUSSIA	8042192159.64
1997	VIETNAM
13744752075.50	1997
RUSSIA	13715752299.36
1996	VIETNAM
13744773115.68	1996
RUSSIA	13758066897.67
1995	VIETNAM
13666298370.50	1995
RUSSIA	13748145021.02
1994	VIETNAM
13717303365.25	1994
RUSSIA	13687724365.74
1993	VIETNAM

1993
13681587159.30
VIETNAM
1992
13749703512.45

175 rows processed.
Statement Processed in 2029.83 seconds.

Ended Executing this Query at Sat Mar 8 01:22:17 1997

Query Started at 857810907.29
Query Ended at 857812937.13
Query Processed in 2029.84 seconds

SQL statements processed: 29
Queries processed: 17

Appendix D. Seed and Query Substitution Parameters

```
-- using 1110820981 as a seed to the RNG
-- Query 1 (Standard)
SELECT L_RETURNFLAG,
       L_LINESTATUS,
       SUM(L_QUANTITY) AS SUM_QTY,
       SUM(L_EXTENDEDPRI) AS SUM_BASE PRICE,
       SUM(L_EXTENDEDPRI * (1 - L_DISCOUNT)) AS
SUM_DISC_PRICE,
       SUM(L_EXTENDEDPRI * (1 - L_DISCOUNT) * (1 + L_TAX)) AS
SUM_CHARGE,
       AVG(L_QUANTITY) AS AVG_QTY,
       AVG(L_EXTENDEDPRI) 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') -
69
GROUP BY L_RETURNFLAG, L_LINESTATUS
ORDER BY L_RETURNFLAG, L_LINESTATUS;
-- 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('1993-11-01','YYYY-MM-DD')
AND O_ORDERDATE < ADD_MONTHS(TO_DATE('1993-11-01','YYYY-
MM-DD'),3)
AND L_COMMITDATE < L_RECEIPTDATE)
GROUP BY O_ORDERPRIORITY
ORDER BY O_ORDERPRIORITY;
-- Query 15 (Variant B)
CREATE TABLE REVENUE0 (SUPPLIER_NO INTEGER, TOTAL_REVENUE
NUMBER(20, 2));

INSERT INTO REVENUE0
SELECT L_SUPPKEY, SUM(L_EXTENDEDPRI*(1-L_DISCOUNT))
FROM LINEITEM
WHERE L_SHIPDATE >= TO_DATE('1993-11-01','YYYY-MM-DD')
AND L_SHIPDATE < ADD_MONTHS(TO_DATE('1993-11-01','YYYY-
MM-DD'),3)
GROUP BY L_SUPPKEY;

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

DROP TABLE REVENUE0;
-- Query 10 (Standard)
set_fetchrows=20;
SELECT
C_CUSTKEY,
```

```
C_NAME,
SUM(L_EXTENDEDPRI * (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('1993-05-01','YYYY-MM-DD')
AND O_ORDERDATE < ADD_MONTHS(TO_DATE('1993-05-01'
,'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;
-- Query 11 (Standard)
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 = 'CANADA'
GROUP BY PS_PARTKEY
HAVING SUM(PS_SUPPLYCOST*PS_AVAILQTY) >
(SELECT
SUM(PS_SUPPLYCOST * PS_AVAILQTY) * 0.0000003333
FROM PARTSUPP, SUPPLIER, NATION
WHERE PS_SUPPKEY = S_SUPPKEY
AND S_NATIONKEY = N_NATIONKEY
AND N_NAME = 'CANADA' )
ORDER BY VALUE DESC;
-- Query 6 (Standard)
SELECT
SUM(L_EXTENDEDPRI * L_DISCOUNT) AS REVENUE
FROM LINEITEM
WHERE L_SHIPDATE >= TO_DATE('1993-01-01','YYYY-MM-DD')
AND L_SHIPDATE < ADD_MONTHS(TO_DATE('1993-01-01','YYYY-
MM-DD'),12)
AND L_DISCOUNT BETWEEN 0.07 - 0.01 AND 0.07 + 0.01
AND L_QUANTITY < 25;
-- Query 2 (Variant A)
set_fetchrows=100;
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 = 8
AND P_TYPE LIKE '%TIN'
AND S_NATIONKEY = N_NATIONKEY
AND N_REGIONKEY = R_REGIONKEY
AND R_NAME = 'EUROPE'
AND (P_PARTKEY ,PS_SUPPLYCOST) IN
(SELECT PS_PARTKEY, MIN(PS_SUPPLYCOST)
FROM PARTSUPP, SUPPLIER, NATION, REGION
WHERE S_SUPPKEY = PS_SUPPKEY
AND S_NATIONKEY = N_NATIONKEY
AND N_REGIONKEY = R_REGIONKEY
AND R_NAME = 'EUROPE'
GROUP BY PS_PARTKEY)
ORDER BY S_ACCTBAL DESC, N_NAME, S_NAME, P_PARTKEY;
-- Query 16 (Standard)
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#14'
AND P_TYPE NOT LIKE 'ECONOMY PLATED%'
AND P_SIZE IN (47, 31 ,49 ,48 ,37, 45, 20, 11)
AND PS_SUPPKEY NOT IN (SELECT S_SUPPKEY
FROM SUPPLIER
```

```

WHERE S_COMMENT LIKE '%Better Business
Bureau%Complaints%'
GROUP BY P_BRAND, P_TYPE, P_SIZE
ORDER BY SUPPLIER_CNT DESC, P_BRAND, P_TYPE, P_SIZE;
-- Query 14 (Variant C)
CREATE TABLE ALL_SALES0 (TYPE VARCHAR2(25), AMOUNT
NUMBER(20, 2));

CREATE TABLE SUM_PROMO_SALES0 (PROMO_AMOUNT NUMBER(20,
2));

CREATE TABLE SUM_ALL_SALES0 (ALL_AMOUNT NUMBER(20, 2));

INSERT INTO ALL_SALES0
SELECT P_TYPE, SUM(L_EXTENDEDPRI * (1-L_DISCOUNT))
FROM LINEITEM, PARTS
WHERE L_PARTKEY = P_PARTKEY
AND L_SHIPDATE >= TO_DATE('1993-11-01', 'YYYY-MM-DD')
AND L_SHIPDATE < ADD_MONTHS(TO_DATE('1993-11-01', 'YYYY-
MM-DD'), 1)
GROUP BY P_TYPE;

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

INSERT INTO SUM_ALL_SALES0
SELECT SUM(AMOUNT)
FROM ALL_SALES0;

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

DROP TABLE ALL_SALES0;

DROP TABLE SUM_PROMO_SALES0;

DROP TABLE SUM_ALL_SALES0;
-- Query 8 (Variant B)
SELECT
YEAR,
SUM(DECODE(NATION, 'CANADA', VOLUME, 0)) / SUM(VOLUME) AS
MKT_SHARE
FROM
(SELECT
TO_CHAR(O_ORDERDATE, 'YYYY') AS YEAR,
L_EXTENDEDPRI * (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 PLATED TIN'
) ALL_NATIONS
GROUP BY YEAR
ORDER BY YEAR;
-- 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 ('AIR', 'FOB')
AND L_COMMITDATE < L_RECEIPTDATE
AND L_SHIPDATE < L_COMMITDATE
AND L_RECEIPTDATE >= TO_DATE('1996-01-01', 'YYYY-MM-DD')
AND L_RECEIPTDATE < ADD_MONTHS(TO_DATE('1996-01-01',
'YYYY-MM-DD'), 12)
GROUP BY L_SHIPMODE
ORDER BY L_SHIPMODE;
-- Query 17 (Variant B)
SELECT SUM(L_EXTENDEDPRI) / 7.0 AS AVG_YEARLY
FROM LINEITEM, (SELECT
P_PARTKEY, 0.2 * AVG(L_QUANTITY) AS AVGQTY
FROM PARTS, LINEITEM
WHERE P_PARTKEY = L_PARTKEY
AND P_BRAND = 'Brand#14'
AND P_CONTAINER = 'JUMBO CAN'
GROUP BY P_PARTKEY) AVG_QUANTITY
WHERE
P_PARTKEY = L_PARTKEY
AND L_QUANTITY < AVGQTY;
-- Query 3 (Standard)
set_fetchrows=10;
SELECT
L_ORDERKEY,
SUM(L_EXTENDEDPRI * (1-L_DISCOUNT)) AS REVENUE,
TO_CHAR(O_ORDERDATE, 'YYYY-MM-DD'),
O_SHIPPRIORITY
FROM CUSTOMER, ORDERS, LINEITEM
WHERE C_MKTSEGMENT = 'AUTOMOBILE'
AND C_CUSTKEY = O_CUSTKEY
AND L_ORDERKEY = O_ORDERKEY
AND O_ORDERDATE < TO_DATE('1995-03-23', 'YYYY-MM-DD')
AND L_SHIPDATE > TO_DATE('1995-03-23', 'YYYY-MM-DD')
GROUP BY L_ORDERKEY, O_ORDERDATE, O_SHIPPRIORITY
ORDER BY REVENUE DESC, O_ORDERDATE;
-- Query 5 (Standard)
SELECT
N_NAME,
SUM(L_EXTENDEDPRI * (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 = 'AFRICA'
AND O_ORDERDATE >= TO_DATE('1996-01-01', 'YYYY-MM-DD')
AND O_ORDERDATE < ADD_MONTHS(TO_DATE('1996-01-01', 'YYYY-
MM-DD'), 12)
GROUP BY N_NAME
ORDER BY REVENUE DESC;
-- Query 13 (Standard)
SELECT
YEAR,
SUM(REVENUE) AS REVENUE
FROM
(SELECT
TO_CHAR(O_ORDERDATE, 'YYYY') AS YEAR,
L_EXTENDEDPRI * (1-L_DISCOUNT) AS REVENUE
FROM LINEITEM, ORDERS
WHERE O_ORDERKEY = L_ORDERKEY
AND O_CLERK = 'Clerk#000000156'
AND L_RETURNFLAG = 'R'
) PERFORMANCE
GROUP BY YEAR
ORDER BY YEAR;
-- Query 7 (Standard)
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_EXTENDEDPRI * (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 = 'CANADA' AND N2.N_NAME = 'CHINA') OR
(N1.N_NAME = 'CHINA' AND N2.N_NAME = 'CANADA'))
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;
-- Query 9 (Standard)
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 '%chiffon%'
) PROFIT
GROUP BY NATION, YEAR
ORDER BY NATION, YEAR DESC;

```

```

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

# The defaults

SF=300
PARAM=256
USER="bench/bench"
QPROG=${SRC_DIR}/qexec.ott
U1PROG=${UPD_SRC}/uf1.ott.reorder
U2PROG=${UPD_SRC}/uf2.ott

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 "          [-sf <scale factor>] [-up <update
parallelism>]"
echo ""
echo "-sf <scale factor> : The scale factor of the run."
echo "                  Default is 300."
echo "-up <update ||ism> : The parallelism to use for the
UFs."
echo "                  Default is 1."
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
dbbench/dbbench."
echo "-h                 : Displays this message."
}

while [ "$#" != "0" ]
do
case $1 in
-sf) shift; SF=$1;;
-up) shift; PARAM=$1;;
-u1) shift; U1PROG=$1;;
-u2) shift; U2PROG=$1;;
-p) shift; QPROG=$1;;
-o) OSTAT=1;;
-s) SSTAT=1;;
-h) usage; exit 0;;
*) echo "Bad argument $1"; usage; exit 1;;
esac
shift;
done

# ${UPD_SPT}/genuf1.sh 1 ${PARAM} ${SF}
# ${UPD_SPT}/genuf1.sh -r 1 ${PARAM} ${SF}
# ${UPD_SPT}/genuf2.sh 1 ${PARAM} ${SF}
${TPCD_BIN}/swtch.sh

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
#RUN_ID=1

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

TPCD_LOG_FILE=${TPCD_LOG}/mp${RUN_ID}qry
TPCD_RPT_FILE=${TPCD_RPT}/mp${RUN_ID}inter
QRY_FILE=${QRY_DIR}/qtemp.${RUN_ID}.${SF}

```

Appendix E. Implementation - Specific Layer/Driver Code

```

#!/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>]
#          [-sf <scale factor>] [-
up <update parallelism>]
#
# Single stream TPC-D Power Test.
#
=====
====#+

TPCD_HOME=${ORACLE_HOME}/tpcd
SCRIPT_DIR=${TPCD_HOME}/scripts
SQL_DIR=${TPCD_HOME}/sql
UPD_DIR=${TPCD_HOME}/update
SRC_DIR=${TPCD_HOME}/source
QRY_DIR=${TPCD_HOME}/queries
QGEN=qgen

DSS_QUERY=${TPCD_HOME}/Squeries
export DSS_QUERY

RUN_ID_FILE=${TPCD_HOME}/r_id

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

```

```

echo "Generates query template file"
${QGEN} -c -p 0 -s ${SF} > ${QRY_FILE}

echo "Done"

if [ $OSTAT ]
then
  ${TPCD_BIN}/start_ora.sh $RUN_ID ${HID}
fi

if [ $SSTAT ]
then
  ${TPCD_BIN}/start_stat.sh ${OUT} ${RUN_ID} ${HID}
fi

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

# Execute UF1

#bstat.sh
echo "Start UF1 at `date`"
UF1_START=`$GTIME`
echo "Start Time for UF1 is $UF1_START"

MAX_DELETE=50
export MAX_DELETE

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

# Execute Query Stream

UF1_END=`$GTIME`
echo "End Time for UF1 is $UF1_END"

#estat.sh

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

${QPROG} $USER q${QRY_FILE} 1${TPCD_LOG_FILE}
r${TPCD_RPT_FILE} \
  > ${OUT}/qs.${RUN_ID}.${HID} 2>&1

# Execute UF2

echo "End Query Stream. Start UF2 at `date`"
UF2_START=`$GTIME`
echo "Start Time for UF2 is $UF2_START"

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

UF2_END=`$GTIME`
echo "End Time for UF2 is $UF2_END"
echo "END UF2 `date`"

END=`$GTIME`

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

UF1_TIME=`echo $UF1_END - $UF1_START | bc`
echo "Elapsed Time for Update Function is $UF1_TIME"
echo "-- Update Function 1" >> ${TPCD_RPT_FILE}
echo "$UF1_TIME" >> ${TPCD_RPT_FILE}

UF2_TIME=`echo $UF2_END - $UF2_START | bc`
echo "Elapsed Time for Update Function is $UF2_TIME"
echo "-- Update Function 2" >> ${TPCD_RPT_FILE}
echo "$UF2_TIME" >> ${TPCD_RPT_FILE}

echo ""
echo "-----"
echo ""

echo "Power Test completed at `date`" >

```

```

${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
${SRC_DIR}/metric ${SF} < ${TPCD_RPT_FILE} \
  >> ${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

THRUPUT=`expr 17 \* 3600 \* ${SF}`
THRUPUT=`echo "scale=2\n${THRUPUT}/${MEA_INT}" | bc`
echo "Real Throughput Metric is: $THRUPUT"
echo "Real Throughput Metric is: $THRUPUT" \
  >> ${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

if [ $OSTAT ]
then
  ${TPCD_BIN}/end_ora.sh $RUN_ID ${HID}
fi

if [ $SSTAT ]
then
  ${TPCD_BIN}/kill_stat.sh ${OUT} ${RUN_ID} ${HID}
fi

#bstat.sh
echo "Restoring UF1 `date`"
RESUF1_START=`$GTIME`
echo "Start Time for RESUF1 is $RESUF1_START"
${UPD_SPT}/resuf1.sh -p ${U2PROG} -u ${USER} ${RUN_ID}
${SF} 1 ${PARA} \
  > ${OUT}/mp${RUN_ID}resuf1 2>&1
RESUF1_END=`$GTIME`
#estat.sh
${TPCD_BIN}/dealloc.sh
echo "Finished Restoring UF1 `date`"
echo "End Time for RESUF1 is $RESUF1_END"
RESUF1_TIME=`echo $RESUF1_END - $RESUF1_START | bc`
echo "Elapsed Time for RESUF1 Function is $RESUF1_TIME"

echo "Restoring UF2 `date`"
RESUF2_START=`$GTIME`
echo "Start Time for RESUF2 is $RESUF2_START"
${UPD_SPT}/resuf2.sh -p ${U1PROG} -u ${USER} ${RUN_ID}
${SF} 1 ${PARA} \
  > ${OUT}/mp${RUN_ID}resuf2 2>&1
#${TPCD_BIN}/swtch.sh
echo "Finished Restoring UF2 `date`"
RESUF2_END=`$GTIME`
echo "End Time for RESUF2 is $RESUF2_END"
RESUF2_TIME=`echo $RESUF2_END - $RESUF2_START | bc`
echo "Elapsed Time for RESUF2 Function is $RESUF2_TIME"

/*=====
=====+
|          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 <errno.h>
#include <math.h>
#include <string.h>
#include <sys/types.h>
#include <time.h>

#include "qexecpl.h"

```



```

        fflush(stderr);
    }

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

        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 = gettimeofday();

    /* 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 statements 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++]);

    /* 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]);
    }

    /* 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] = (dtype *) memalloc(sizeof(dtype));

        /* 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]->rflen, (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 i,j;
    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)) {
        OEXPET(&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)
    lda def *lda;
    csr def *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, diff;
    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 diff;
    int cwid;

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

        len = 0;

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

```

```

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

        /* do a little bit of formatting */

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

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

/* remove_newline(): Remove newline character from str. */
void remove_newline(str)
    char *str;
{
    char *p;

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

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

#ifdef 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

#ifdef TRUE
#define TRUE 1
#endif /* TRUE */

#ifdef 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 */

list. /* select
*/

/* 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,buf1,ftype,scale,indp,fmt,fmt1,fmt
t,rlen,rcode)\
    if
(odefin((cursor),(pos),(ub1*)(buf),(buf1),(ftype),(scale),\
(sb2
*) (indp),(text*)(fmt),(fmt1),(fmtt),(rlen),(rcode)))\
        {sql_error(lda,cursor);}\
    else\
        DISCARD 0

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

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

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

#define ISOTXT "alter session set isolation_level =
serializable"

#endif /* QSTREAMPL_H */

#!/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=$ORACLE_HOME
TPCD_DIR=${O}/tpcd
UPDATE_DIR=${TPCD_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
SRC_DIR=${UPDATE_DIR}/source
GTIME_DIR=${TPCD_DIR}/source

```

```

GTIME=${GTIME_DIR}/gtime

usage() {
echo ""
echo "runufl.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.ott"
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.ott
LOGPATH=.
PASSWD="dbbench/dbbench"
MAXPROC=63
PROCS="`count 0 $MAXPROC`"
PROCS="$PROCS $PROCS $PROCS $PROCS"

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

i=1
PID=""

# perform the update function 1

START=`GTIME`

set $PROCS
for i in `count 1 $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.

PROC=$1
pbind -b $PROC $$ >/dev/null
${PROG} ${SETNUM} ${RUN_ID} ${i} ${SF} f${LOGPATH}
u${PASSWD} &
shift

done

pbind -u $$

# All update processes have 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 ""

/*
* Sccsid:      @(#)shared.h9.1.1.2      1/30/95  12:52:13
*
*/
#define N_CMNT_LEN      72
#define N_CMNT_MAX      152
#define MONEY_SCL      0.01
#define V_STR_HGH      1.6
#define P_NAME_LEN      55
#define P_MFG_LEN      25
#define P_BRND_LEN      10
#define P_TYPE_LEN      25
#define P_CNTR_LEN      10
#define P_CMNT_LEN      14
#define P_CMNT_MAX      23
#define S_NAME_LEN      25
#define S_ADDR_LEN      25
#define S_ADDR_MAX      40
#define S_CMNT_LEN      63
#define S_CMNT_MAX      101
#define PS_CMNT_LEN      124
#define PS_CMNT_MAX      199
#define C_NAME_LEN      25
#define C_ADDR_LEN      25
#define C_ADDR_MAX      40
#define C_MSEG_LEN      10
#define C_CMNT_LEN      73
#define C_CMNT_MAX      117
#define O_OPRIO_LEN      15
#define O_CLRK_LEN      15
#define O_CMNT_LEN      49
#define O_CMNT_MAX      79
#define L_CMNT_LEN      27
#define L_CMNT_MAX      44
#define L_INST_LEN      25
#define L_SMODE_LEN      10
#define T_ALPHA_LEN      10
#define DATE_LEN      13 /* long enough to hold either
date format */
#define NATION_LEN      25
#define REGION_LEN      25
#define PHONE_LEN      15
#define MAXAGG_LEN      20 /* max component length for a
agg str */
#define P_CMNT_SD      6
#define PS_CMNT_SD      9
#define O_CMNT_SD      12
#define L_CMNT_SD      19
#define C_ADDR_SD      26
#define C_CMNT_SD      31
#define S_ADDR_SD      32
#define S_CMNT_SD      36

#ifdef RCSID
static char *RCSid =
"$Header: testtime.c 7030100.1 96/05/24 12:56:30 plai
Generic<base> $ Copyr (c) 1995 Oracle";
#endif /* RCSID */

/*=====

```

```

=====+
|           Copyright (c) 1996  Oracle Corp, Redwood Shores,
CA           |
|           OPEN SYSTEMS PERFORMANCE
GROUP        |
|           All Rights
Reserved     |
+-----+
| FILENAME
|   testtime.c
| DESCRIPTION
|   Test gettime.c.
+-----+
+-----*/
#include <stdio.h>

main ()
{
    double gettime ();
    double t1, t2;

    t1 = gettime ();
    sleep (30);
    t2 = gettime ();

    printf("Time Start is %f Time End is %f\n",t1, t2);
    printf ("Slept 30 sec.  Gettime () reported %f sec.\n",
t2 - t1);
    printf ("If the numbers don't match, you need to
modify\n");
    printf ("bench/gen/source/gettime.c to fix this\n");

    exit (0);
}

/*=====+
|           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 "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>

/* Include 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[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_okekey[MAX_ORD_INSERTS];
long o_ckekey[MAX_ORD_INSERTS];
varchar o_ostat[MAX_ORD_INSERTS][2];
float o_tprice[MAX_ORD_INSERTS];
varchar o_odate[MAX_ORD_INSERTS][DATE_LEN];
varchar o_opri[MAX_ORD_INSERTS][O_OPRIO_LEN+1];
varchar o_clk[MAX_ORD_INSERTS][O_CLRK_LEN+1];
int o_spri[MAX_ORD_INSERTS];
varchar o_cmnt[MAX_ORD_INSERTS][O_CMNT_MAX+1];

/* declarations for LINEITEM */

long l_okekey[MAX_LINE_INSERTS];
long l_pkey[MAX_LINE_INSERTS];
long l_skey[MAX_LINE_INSERTS];
int l_lnum[MAX_LINE_INSERTS];
float l_quan[MAX_LINE_INSERTS];
float l_eprice[MAX_LINE_INSERTS];
float l_disc[MAX_LINE_INSERTS];
float l_tax[MAX_LINE_INSERTS];
varchar l_rflag[MAX_LINE_INSERTS][2];
varchar l_lstat[MAX_LINE_INSERTS][2];
varchar l_sdate[MAX_LINE_INSERTS][DATE_LEN];
varchar l_cdate[MAX_LINE_INSERTS][DATE_LEN];
varchar l_rdate[MAX_LINE_INSERTS][DATE_LEN];
varchar l_sinst[MAX_LINE_INSERTS][L_INST_LEN+1];
varchar l_smode[MAX_LINE_INSERTS][L_SMODE_LEN+1];
varchar l_cmnt[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: ufl.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");
    fprintf(stderr, "    r              :indicates
that this is a recovery run\n");
    fprintf(stderr, "    f<path name for log> :full path
name for reports\n");
    fprintf(stderr, "    u<uid/passwd>
:Username/Password string - default is tpcd/tpcd\n");
    exit(-1);
}

void sql_error()
{
    int i;

    /* 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 i;
    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];

    /* 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 $ORACLE_HOME/update/data. */

    if (!recover) {
        sprintf(ordpath, "%s/tpcd/update/data/order.tbl.u%d.d",
getenv("ORACLE_HOME"), set_id,
proc_no);
        sprintf(itempath,
"%s/tpcd/update/data/lineitem.tbl.u%d.d",
getenv("ORACLE_HOME"), set_id,
proc_no);

```



```

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 */
        /* invocation 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_guan[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_cmmt[i].arr, pos1);
        l_cmmt[i].len = strlen((char *)
l_cmmt[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;
}

#!/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=$ORACLE_HOME
TPCD_DIR=$O/tpcd
UPDATE_DIR=${TPCD_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
SRC_DIR=${UPDATE_DIR}/source
GTIME_DIR=${TPCD_DIR}/source
GTIME=${GTIME_DIR}/gtime

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.ott."
        echo "-u : Userid/Password for Oracle. Default is
dbbench/dbbench"
        echo "-h : To Display this message."
        echo ""
    }

    PROG=${SRC_DIR}/uf2.ott
    PASSWD="dbbench/dbbench"
    MAXPROC=63
    PROCS="`count 0 $MAXPROC`"
    PROCS="$PROCS $PROCS $PROCS $PROCS"

    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

    i=1
    PID=""

    START=`$GTIME`

    # perform the update function

    set $PROCS
    for i in `count 1 $PAR`
    do

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

        PROC=$1
        pbind -b $PROC $$ >/dev/null
        ${PROG} ${SETNUM} ${RUN_ID} ${i} ${SF} 1 u${PASSWD} &
        shift
    done

    pbind -u $$

    # All processes have 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 ""

    /*=====
=====

```

```

|      Copyright (c) 1995 Oracle Corp, Redwood Shores,
CA      |
|      OPEN SYSTEMS PERFORMANCE
GROUP   |
|      All Rights
Reserved|

+=====
=====+
| FILENAME
|   ufl_del.pc
| DESCRIPTION
|   TPC-D benchmark uf2 driver, PRO*C version
|   Also used to restore ufl.
| 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 "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 :

|      Copyright (c) 1995 Oracle Corp, Redwood Shores,
CA      |
|      OPEN SYSTEMS PERFORMANCE
GROUP   |
|      All Rights
Reserved|

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>

/* Include 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();

#ifdef NULL
#define NULL 0
#endif

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

EXEC SQL BEGIN DECLARE SECTION;

varchar sqlstmt[SQL_LEN]; /* the delete statement
for orders */
varchar sqlstmt2[SQL_LEN]; /* the delete statement
for lineitem */
VARCHAR lname[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_del.o[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");
    fprintf(stderr, "    usage flag     :enter 1
for uf2 and 2 for restoring ufl\n");
    fprintf(stderr, "    u<uid/passwd>
:Username/Password string - default is tpcd/tpcd\n");
    exit(-1);
}

```

```

void sql_error()
{
    int i;

    /* 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 i;
    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 */

    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 $ORACLE_HOME/update/data. */

    u_flag = atoi(argv[5]);

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

    /* 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) {
            EXEC SQL EXECUTE IMMEDIATE :sqlstmt;
            o = sqlca.sqlerrd[2];
            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 */

/* 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];

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

        /* extract columns from the line */

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

        if (i == 0) {
            /* prepare statement for ORDERS and LINEITEM */
            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
*/
            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);
}
}
}
return (MAX_INSERTS);
}

void wakeup() {
    return;
}

```

Appendix F. Database Layout

Db Entity	Volume
Lineitem	sf300_li_<xxx>
Orders	sf300_o_<xx>
Partsupp	sf300_ps_<xx>
Parts	sf300_pt_<xx>
Supplier	sf300_su_<xx>
Customer	sf300_c_<xx>
All Indexes	sf300_ind_<xx>
Temp space	sf300_temp_<xx>
User temp	sf300_user_0
Rollback	sf300_rbs_<x>
System file	sf300_sys_0
Log files	sf300_redo_<x>

The verification database used similar naming convention except for the file prefix of 01 instead of sf300.

The complete veritas listing describing the layout takes over 320 pages. Rather than including that, we are including the scripts and programs that we used to generated the commands that were used to generate the volumes.

Our basic philosophy was to strip all the main tablespaces across all of the disks. With each database file (that composed the tablespaces) was striped 23 ways and mirror on different disks. Expect for temp and index which were not mirror. Each tablespace was comprised of 64 of the database files that had been striped in this way. Lineitem and temp were composed of 128 files.

Veritas commands are generated by a couple of simple C programs.

```

-----
README
-----
I. To set up the disk volumes:

    Run gen_disks.sh and redirect output to gen_vx.sh.

    chmod and run gen_vx.sh and redirect output to
    vx.cmds.sh.

    (As root)

    chmod and run vx.cmds.sh.

-----
gen_disks.sh
-----
#!/bin/sh
tpcdmake_sf -s 300 -f 64 -n 23 -r 16 -l sae1.vx61.disks -m
-----

```

```

gen_vx.sh
-----

tpcddisklist -n 5888 -s 0 -r 16 < sael.vx61.disks | \
tpcddisks -f 128 -n 23 -t li -s 234000000000 -S 300 -m
tpcddisklist -n 8832 -s 5888 -r 16 < sael.vx61.disks | \
tpcddisks -f 64 -n 23 -t ps -s 384000000000 -S 300 -m
tpcddisklist -n 11776 -s 8832 -r 16 < sael.vx61.disks | \
tpcddisks -f 64 -n 23 -t o -s 549000000000 -S 300 -m
tpcddisklist -n 13248 -s 11776 -r 16 < sael.vx61.disks | \
tpcddisks -f 64 -n 23 -t ind -s 270000000000 -S 300
tpcddisklist -n 16192 -s 13248 -r 16 < sael.vx61.disks | \
tpcddisks -f 128 -n 23 -t temp -s 234000000000 -S 300
tpcddisklist -n 19136 -s 16192 -r 16 < sael.vx61.disks | \
tpcddisks -f 64 -n 23 -t pt -s 9000000000 -S 300 -m
tpcddisklist -n 22080 -s 19136 -r 16 < sael.vx61.disks | \
tpcddisks -f 64 -n 23 -t c -s 8100000000 -S 300 -m
tpcddisklist -n 25024 -s 22080 -r 16 < sael.vx61.disks | \
tpcddisks -f 64 -n 23 -t su -s 4800000000 -S 300 -m
tpcddisklist -n 25028 -s 25024 -r 16 < sael.vx61.disks | \
tpcddisks -f 1 -n 2 -t sys -s 51380224 -S 300 -m
tpcddisklist -n 25124 -s 25028 -r 16 < sael.vx61.disks | \
tpcddisks -f 3 -n 16 -t rbs -s 558891008 -S 300 -m
tpcddisklist -n 25140 -s 25124 -r 16 < sael.vx61.disks | \
tpcddisks -f 1 -n 16 -t user -s 504365056 -S 300
tpcddisklist -n 25146 -s 25140 -r 16 < sael.vx61.disks | \
tpcddisks -f 3 -n 1 -t redo -s 2097152000 -S 300 -m

-----
vx.cmds.sh
-----

(This is a sample of the lines generated here).

echo Starting li at `date`

(
vxedit -fr rm sf300_li_0
vxassist -g rootdg -U gen make sf300_li_0 1964928K \
  layout=stripe nstripe=23 stripe_width=64K user=oracle
group=dba \
c1t0d0 c2t0d0 c3t0d0 c4t0d0 c5t0d0 c6t0d0 c7t0d0 c8t0d0
c9t0d0 \
c10t0d0 c11t0d0 c12t0d0 c13t0d0 c14t0d0 c15t0d0 c16t0d0
c17t0d0 c18t0d0 \
c19t0d0 c20t0d0 c21t0d0 c22t0d0 c23t0d0
vxassist -g rootdg -U gen -o slow=1,iosize=2048 mirror
sf300_li_0 \
  layout=stripe nstripe=23 stripe_width=64K \
c24t0d0 c25t0d0 c26t0d0 c27t0d0 c28t0d0 c29t0d0 c30t0d0
c31t0d0 c32t0d0 \
c33t0d0 c34t0d0 c35t0d0 c36t0d0 c37t0d0 c38t0d0 c39t0d0
c40t0d0 c41t0d0 \
c42t0d0 c43t0d0 c44t0d0 c45t0d0 c46t0d0
)&

(
vxedit -fr rm sf300_li_1
vxassist -g rootdg -U gen make sf300_li_1 1964928K \
  layout=stripe nstripe=23 stripe_width=64K user=oracle
group=dba \
c48t0d0 c49t0d0 c50t0d0 c51t0d0 c52t0d0 c53t0d0 c54t0d0
c55t0d0 c56t0d0 \
c57t0d0 c58t0d0 c59t0d0 c60t0d0 c61t0d0 c17t1d0 c18t1d0
c19t1d0 c20t1d0 \
c21t1d0 c22t1d0 c23t1d0 c24t1d0 c25t1d0
vxassist -g rootdg -U gen -o slow=1,iosize=2048 mirror
sf300_li_1 \
  layout=stripe nstripe=23 stripe_width=64K \
c26t1d0 c27t1d0 c28t1d0 c29t1d0 c30t1d0 c31t1d0 c32t1d0
c33t1d0 c34t1d0 \
c35t1d0 c36t1d0 c37t1d0 c38t1d0 c39t1d0 c40t1d0 c41t1d0
c42t1d0 c43t1d0 \
c44t1d0 c45t1d0 c46t1d0 c48t1d0 c49t1d0
)&

...

(repeats for each file for each tablespace, rotating
through all the disks.)

wait

echo Starting rbs at `date`

(
vxedit -fr rm sf300_rbs_0
vxassist -g rootdg -U gen make sf300_rbs_0 545280K \
  layout=stripe nstripe=16 stripe_width=64K user=oracle
group=dba \
c21t4d0 c22t4d0 c23t4d0 c24t4d0 c25t4d0 c26t4d0 c27t4d0
c28t4d0 c29t4d0 \
c30t4d0 c31t4d0 c32t4d0 c33t4d0 c34t4d0 c35t4d0 c36t4d0
vxassist -g rootdg -U gen -o slow=1,iosize=2048 mirror
sf300_rbs_0 \
  layout=stripe nstripe=16 stripe_width=64K \
c37t4d0 c38t4d0 c39t4d0 c40t4d0 c41t4d0 c42t4d0 c43t4d0
c44t4d0 c45t4d0 \
c46t4d0 c48t4d0 c49t4d0 c50t4d0 c51t4d0 c52t4d0 c53t4d0
)&

(
vxedit -fr rm sf300_rbs_1
vxassist -g rootdg -U gen make sf300_rbs_1 545280K \
  layout=stripe nstripe=16 stripe_width=64K user=oracle
group=dba \
c54t4d0 c55t4d0 c56t4d0 c57t4d0 c58t4d0 c59t4d0 c60t4d0
c61t4d0 c1t4d0 \
c2t4d0 c3t4d0 c4t4d0 c5t4d0 c6t4d0 c7t4d0 c8t4d0
vxassist -g rootdg -U gen -o slow=1,iosize=2048 mirror
sf300_rbs_1 \
  layout=stripe nstripe=16 stripe_width=64K \
c9t4d0 c10t4d0 c11t4d0 c12t4d0 c29t5d0 c30t5d0 c31t5d0
c32t5d0 c33t5d0 \
c34t5d0 c35t5d0 c36t5d0 c37t5d0 c38t5d0 c39t5d0 c40t5d0
)&

(
vxedit -fr rm sf300_rbs_2
vxassist -g rootdg -U gen make sf300_rbs_2 545280K \
  layout=stripe nstripe=16 stripe_width=64K user=oracle
group=dba \
c41t5d0 c42t5d0 c43t5d0 c44t5d0 c45t5d0 c46t5d0 c48t5d0
c49t5d0 c50t5d0 \
c51t5d0 c52t5d0 c53t5d0 c54t5d0 c55t5d0 c56t5d0 c57t5d0
vxassist -g rootdg -U gen -o slow=1,iosize=2048 mirror
sf300_rbs_2 \
  layout=stripe nstripe=16 stripe_width=64K \
c58t5d0 c59t5d0 c60t5d0 c61t5d0 c1t5d0 c2t5d0 c3t5d0 c4t5d0
c5t5d0 \
c6t5d0 c7t5d0 c8t5d0 c9t5d0 c10t5d0 c11t5d0 c12t5d0
)&
echo Done with rbs at `date`
echo Starting redo at `date`

(
vxedit -fr rm sf300_redo_0
vxassist -g valid -U gen make sf300_redo_0 2000M
user=oracle group=dba \
c47t0d0
vxassist -g valid -U gen -o slow=1,iosize=2048 mirror
sf300_redo_0 \
c47t1d0
)&

(
vxedit -fr rm sf300_redo_1
vxassist -g valid -U gen make sf300_redo_1 2000M
user=oracle group=dba \
c47t0d0
vxassist -g valid -U gen -o slow=1,iosize=2048 mirror
sf300_redo_1 \
c47t1d0
)&

(
vxedit -fr rm sf300_redo_2
vxassist -g valid -U gen make sf300_redo_2 2000M
user=oracle group=dba \
c47t2d0
vxassist -g valid -U gen -o slow=1,iosize=2048 mirror
sf300_redo_2 \
c47t3d0
)&
wait
echo Done with redo at `date`

```

```
echo All done at `date`
```

```
-----  
tpcdmake_sf.c  
-----
```

```
/*  
*****  
******/  
/*  
/* Program Name : tpcdmake_sf.c  
/*  
/* Description :  
/* This program takes a scale factor and generates the  
script to build*/  
/*  
/* Calling Sequence :  
/*  
/* Inputs :  
/*  
/* Outputs :  
/*  
/* Revision History  
/* Version - Author - Date - Reason  
/* 1.00.00 djm 10-Oct-96 Initial Edit  
/*  
*****  
******/
```

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

```
#define SYS_SIZE (49*1024*1024)  
#define REDO_SIZE (2000*1024*1024)  
#define RBS_SIZE (533*1024*1024)  
#define USER_SIZE (481*1024*1024)
```

```
struct _tables  
{  
    char table_name[20];  
    char ts_abbrev[10];  
    int rows_per_sf;  
    int row_size;  
    char mirror_st[5];  
    int file_count;  
    int nstripe;  
    int file_mult;  
} tables [] =  
{  
    { "lineitem", "li", 6000000, 130, "-m", 0, 0, 2 },  
    { "partsupp", "ps", 800000, 160, "-m", 0, 0, 1 },  
    { "orders", "o", 1500000, 122, "-m", 0, 0, 1 },  
    { "indexes", "ind", 6000000, 150, "", 0, 0, 1 },  
    { "temp", "temp", 6000000, 130, "", 0, 0, 2 },  
    { "parts", "pt", 200000, 150, "-m", 0, 0, 1 },  
    { "customer", "c", 150000, 180, "-m", 0, 0, 1 },  
    { "supplier", "su", 10000, 160, "-m", 0, 0, 1 },  
    { "system", "sys", SYS_SIZE, 1, "-m", 1, 2, 1 },  
    { "rbs", "rbs", RBS_SIZE, 1, "-m", 3, 16, 1 },  
    { "user", "user", USER_SIZE, 1, "", 1, 16, 1 },  
    { "redo", "redo", REDO_SIZE, 1, "-m", 3, 1, 1 }  
};
```

```
main(argc, argv)  
    int argc;  
    char *argv[];  
{  
    int ch;  
    int i;  
    int n_files;  
    int scale_factor = 0;  
    int files_per_ts = 0;  
    int fp_ts;  
    int nstripe = 0;  
    int rotate = 0;  
    int skip_count = 0;  
    int mirror = 0;  
    int do_sed = 0;  
    int split = 0;  
    char *mirror_st;  
    char no_mirror[] = " ";  
    char do_mirror[] = "-m";  
    char disk_file[100];  
    long long size = 0;  
    extern char *optarg;  
    extern int optind;
```

```
while ((ch = getopt(argc, argv, "d:f:l:mn:r:s:S")) !=  
EOF)  
    switch (ch)  
    {  
        case 'd' :  
            sscanf(optarg, "%d",  
&split);  
            break;  
        case 'f' :  
            sscanf(optarg, "%d",  
&files_per_ts);  
            break;  
        case 'l' :  
            strcpy(disk_file,  
optarg);  
            break;  
        case 'm' :  
            mirror = 1;  
            break;  
        case 'n' :  
            sscanf(optarg, "%d",  
&nstripe);  
            break;  
        case 'r' :  
            sscanf(optarg, "%d", &rotate);  
            break;  
        case 's' :  
            sscanf(optarg, "%d",  
&scale_factor);  
            break;  
        case 'S' :  
            do_sed = 1;  
            break;  
        case '?' :  
            printf("usage:  
tpcdmake_sf -s scale_factor -f files_per_ts - n nstripe -r  
rotate -l disk_list -d dbgen_split [-m]\n");  
            exit(1);  
    }  
  
    for (i = 0; i < sizeof(tables)/sizeof(struct _tables);  
i++)  
    {  
        /* for special files, don't need  
scale_factor */  
        if (tables[i].file_count == 0)  
        {  
            size = tables[i].row_size *  
scale_factor;  
            fp_ts = files_per_ts *  
tables[i].file_mult;  
        }  
        else  
        {  
            /* adjust the nstripe and files  
appropriately */  
            size = tables[i].row_size;  
            nstripe = tables[i].nstripe;  
            fp_ts = tables[i].file_count *  
tables[i].file_mult;  
            n_files = fp_ts*nstripe;  
            size = size * tables[i].rows_per_sf;  
            if (mirror)  
            {  
                if (strcmp(tables[i].mirror_st, "-m")  
== 0)  
                    n_files *= 2;  
                mirror_st = tables[i].mirror_st;  
            }  
            else  
                mirror_st = no_mirror;  
            if (do_sed)  
            {  
                printf("tpcdgen_sf -f %d -n %d -t %s -  
s %lld -S %d -d %d %s\n",  
                    fp_ts, nstripe,
```

```

tables[i].ts_abbrev, size, scale_factor,
        split, mirror_st);
    }
    else
    {
        printf("tpcddisklist -n %d -s %d -r %d
        n_files + skip_count,
        skip_count, rotate,
disk_file);
        printf("tpcddisks -f %d -n %d -t %s -s
        fp_ts, nstripe,
tables[i].ts_abbrev, size, scale_factor,
        mirror_st);
        skip_count += n_files;
    }
}

-----
tpcddisklist.c
-----
/*****
*****
/*
/* Program Name : tpcddisklist.c
/*
/* Description :
/* This program reads output from "format" and generates
disk list */
/*
/* Calling Sequence :
/*
/* Inputs :
/*
/* Outputs :
/*
/* Revision History
/* Version - Author - Date - Reason
/* 1.00.00 djm 10-Oct-96 Initial Edit
/*****
*****

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

struct d_list_type
{
    int d;
    struct d_list_type *d_next_ptr;
};

struct t_list_type
{
    int t;
    struct d_list_type *d_list_ptr;
    struct d_list_type *d_curr_ptr;
    struct t_list_type *t_next_ptr;
    int t_done;
};

struct c_list_type
{
    int c;
    struct t_list_type *t_list_ptr;
    struct t_list_type *t_curr_ptr;
    struct c_list_type *c_next_ptr;
    int c_done;
} c_list = {0}, *c_last = &c_list;

long long getsize(disktype)
char *disktype;
{
/*
# ST43401N 3GB
# ST43402ND 3GB
# SUN1.05G 1.05GB
# ZITEL 4GB
# SUN9.0G 9GB
# SUN2.1G 2.1GB
*/

long long GB = 1024 * 1024;
GB = GB * 1024;

if (strcmp(disktype, "SUN1.05G") == 0)
{
    return (long long) (1.05*GB);
}
if (strncmp(disktype, "SEAGATE-ST43401N", 16) == 0)
{
    return (long long) (3*GB);
}
if (strncmp(disktype, "SEAGATE-ST43402ND", 18) == 0)
{
    return (long long) (3*GB);
}
if (strcmp(disktype, "ZITEL") == 0)
{
    return (long long) (4*GB);
}
if (strcmp(disktype, "SUN9.0G") == 0)
{
    return (long long) (9*GB);
}
if (strcmp(disktype, "SUN2.1G") == 0)
{
    return (long long) (2.1*GB);
}
printf("bad disk type %s\n", disktype);
return (long long) 0;
}

add_disk(c,t,d)
int c;
int t;
int d;
{
    struct c_list_type *c_ptr;
    struct t_list_type *t_ptr, *t_last;
    struct d_list_type *d_ptr, *d_last;

    for (c_ptr = c_list.c_next_ptr; c_ptr != NULL; c_ptr =
c_ptr->c_next_ptr)
    {
        if (c_ptr->c == c)
            break;
    }

    if (c_ptr == NULL)
    {
        c_ptr = malloc(sizeof(struct c_list_type));
        c_ptr->c = c;
        c_ptr->c_next_ptr = NULL;
        c_ptr->t_curr_ptr = NULL;
        c_ptr->d_curr_ptr = NULL;
        c_ptr->t_list_ptr = NULL;
        c_ptr->d_done = 0;

        t_ptr = NULL;

        for (t_ptr = c_ptr->t_list_ptr; t_ptr != NULL; t_ptr =
t_ptr->t_next_ptr)
        {
            if (t_ptr->t == t)
                break;
            t_ptr = t_ptr->t_next_ptr;
        }

        if (t_ptr == NULL)
        {
            t_ptr = malloc(sizeof(struct t_list_type));
            t_ptr->t = t;
            t_ptr->t_next_ptr = NULL;
            t_ptr->d_curr_ptr = NULL;
            if (t_ptr->t_list_ptr == NULL)
                c_ptr->t_list_ptr = t_ptr;
            else
                t_ptr->t_next_ptr = t_ptr;
            t_ptr->d_list_ptr = NULL;
            t_ptr->d_done = 0;
        }
    }
}

```

```

d_last = NULL;
for (d_ptr = t_ptr->d_list_ptr; d_ptr != NULL; d_ptr =
d_ptr->d_next_ptr)
{
    if (d_ptr->d == d)
        break;
    d_last = d_ptr;
}
if (d_ptr == NULL)
{
    d_ptr = malloc(sizeof(struct d_list_type));
    d_ptr->d = d;
    d_ptr->d_next_ptr = NULL;
    if (d_last == NULL)
        t_ptr->d_list_ptr = d_ptr;
    else
        d_last->d_next_ptr = d_ptr;
}
}
check_done(c_ptr)
struct c_list_type *c_ptr;
{
    struct t_list_type *t_ptr;
    for (t_ptr = c_ptr->t_list_ptr; t_ptr != NULL; t_ptr =
t_ptr->t_next_ptr)
        if (t_ptr->t_done != 1)
            return(0);
    return(1);
}
rotate_c_list(rotate)
int rotate;
{
    struct c_list_type *c_ptr, *c_prev;
    int i;
    c_ptr = c_list.c_next_ptr;
    for (i = 0; i < rotate; i++)
    {
        c_prev = c_ptr;
        if (c_ptr == NULL)
            c_ptr = c_list.c_next_ptr;
        else
            c_ptr = c_ptr->c_next_ptr;
    }
    c_prev->c_next_ptr = NULL;
    c_last->c_next_ptr = c_list.c_next_ptr;
    c_list.c_next_ptr = c_ptr;
    c_last = c_prev;
}
print_nodes(numdisks, skipdisks, rotate)
int numdisks;
int skipdisks;
int rotate;
{
    struct c_list_type *c_ptr;
    struct t_list_type *t_ptr;
    struct d_list_type *d_ptr;
    int all_c_done = 0;
    int count_disks = 0;
    while (! all_c_done)
    {
        all_c_done = 1;
        for (c_ptr = c_list.c_next_ptr; c_ptr !=
NULL; c_ptr = c_ptr->c_next_ptr)
        {
            if (c_ptr->t_curr_ptr == NULL)
                t_ptr = c_ptr-
>t_list_ptr;
            else if (c_ptr->t_curr_ptr->t_next_ptr
== NULL)
                t_ptr = c_ptr-
>t_list_ptr;
            else
                t_ptr = c_ptr-
>t_curr_ptr->t_next_ptr;
            c_ptr->t_curr_ptr = t_ptr;
            if (t_ptr->d_curr_ptr == NULL)
                d_ptr = t_ptr-
>d_list_ptr;
            else if (t_ptr->d_curr_ptr->d_next_ptr
== NULL)
                d_ptr = t_ptr-
>d_list_ptr;
            else
                d_ptr = t_ptr-
>d_curr_ptr->d_next_ptr;
            if (d_ptr->d_next_ptr == NULL)
                t_ptr->t_done = 1;
            t_ptr->d_curr_ptr = d_ptr;
            if (count_disks >= skipdisks)
                printf("c%dt%dd%dn",
c_ptr->c, t_ptr->t, d_ptr->d);
            all_c_done &= check_done(c_ptr);
            count_disks++;
            if (numdisks > 0)
            {
                if (count_disks >=
numdisks)
                {
                    all_c_done = 1;
                    break;
                }
                else
                    all_c_done = 0;
            }
        }
    }
    if (rotate)
        rotate_c_list(rotate);
}
main(argc, argv)
int argc;
char *argv[];
{
    int ch;
    int cc, tt, dd;
    int c = 0;
    int t = 0;
    int d = 0;
    int numdisks = 0;
    int skipdisks = 0;
    int rotate = 0;
    int oldc, oldt, oldd;
    long long size;
    extern char *optarg;
    extern int optind;
    char instring[132];
    char dunc, diskname[20], disktype[40];
    float dumf;
    while ((ch = getopt(argc, argv, "n:r:s:")) != EOF)
        switch (ch)
        {
            case 'n' :
                sscanf(optarg, "%d",
&numdisks);
                break;
            case 'r' :
                sscanf(optarg, "%d",
&rotate);
                break;
            case 's' :
                sscanf(optarg, "%d",
&skipdisks);
                break;
            case '?' :
                printf("usage:

```

```

tpcddisklist -n numdisks -s skipdisks -r
rotate_btwn_ctrl\n");
                                exit(1);
                                }
                                while (gets(instrstring))
                                {
                                sscanf(instrstring, "%f %s %c %s", &dumf,
diskname, &dumc, disktype);
                                sscanf(diskname, "c%dt%dd%d", &c, &t, &d);
                                size = getsize(disktype);
                                /*
                                printf("%s, %d, %d, %d, %s, %lld\n",
diskname, c, t, d, disktype, size);
                                */
                                add_disk(c,t,d);
                                }
                                print_nodes(numdisks, skipdisks, rotate);
                                }

-----
tpcddisks.c
-----
/*****
*****/
/*
/* Program Name : tpcddisks.c
/*
/* Description :
/* This program configures tpcd disks and
tables
/*
/* Calling Sequence :
/*
/* Inputs :
/*
/* Outputs :
/*
/* Revision History
/* Version - Author - Date - Reason
/* 1.00.00 djm 22-Oct-96 Initial Edit
/*****
*****/

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

/* Overhead of 10% extra per file */
#define OVERHEAD 1.1
#define STRIPE_WIDTH (64*1024)
#define ROUND (64*1024)
#define MAX_BEFORE_WAIT 54
#define FIRST_SIZE 1146880

main(argc, argv)
int argc;
char *argv[];
{
int ch;
int f = 0;
int nstripe = 0;
int sf = 0;
int ncpus = 0;
int i,j,m;
int before_wait;
int max_before_wait = MAX_BEFORE_WAIT;
long long size = 0;
long long perfilesize = 0;
char name[20];
char diskname[20];
char volname[20];
char volstring[1024];
int mult;
int mirror = 0;
extern char *optarg;
extern int optind;

while ((ch = getopt(argc, argv, "c:f:mn:t:s:S:")) !=
EOF)
switch (ch)
{
case 'c' :
sscanf(optarg, "%d",
&ncpus);
break;
case 'f' :
sscanf(optarg, "%d", &f);
break;
case 'm' :
mirror = 1;
break;
case 'n' :
sscanf(optarg, "%d",
&nstripe);
break;
case 't' :
strcpy(name, optarg);
break;
case 'S' :
sscanf(optarg, "%d",
&sf);
break;
case 's' :
if
(optarg[strlen(optarg)-1] == 'm' ||
optarg[strlen(optarg)-1] == 'M')
mult = 1024*1024;
else if
(optarg[strlen(optarg)-1] == 'k' ||
optarg[strlen(optarg)-1] == 'K')
mult = 1024;
else
mult = 1;
sscanf(optarg, "%lld",
&size);
size = size * mult;
break;
case '?' :
printf("usage: tpcddisks
-f nfiles -n nstripe -t tablespacename -s size -S
scale_factor [-m] [-c cpus]\n");
exit(1);
}

if (f <= 0 || nstripe <= 0 || size <= 0 || sf <= 0)
{
printf("usage: tpcddisks -f nfiles -n
nstripe -t tablespacename -s size -S scale_factor [-m] [-c
cpus]\n");
exit(1);
}

/* get perfilesize, including overhead */
perfilesize = size/f * OVERHEAD + FIRST_SIZE;

/* now round up to even 64K block */
perfilesize = ((perfilesize + (ROUND - 1))/ROUND) *
ROUND;

if (ncpus > 0)
max_before_wait = ncpus;

before_wait = 0;

printf("echo Starting %s at `date`\n", name);

for (i = 0; i < f ; i++)
{
if (before_wait++ >= max_before_wait)
{
printf("wait\n");
before_wait = 0;
}

for (m = 0; m <= mirror; m++)
{
volstring[0] = NULL;
for (j = 0; j < nstripe; j++)
{
gets(diskname);
/*
printf("insert into

```

```

DISK_USED \n"
"values
( '%s', '%s', %lld, %d);\n",
name,
diskname, perfilesize/nstripe, i);
*/
if ((j % 9) == 0)
    strcat(volstring,
"\n");
    strcat(volstring,
diskname);
    strcat(volstring, " ");
}
sprintf(volname, "sf%d_%s%d", sf,
name, i);
if (!m)
{
    printf("\n\n");
    printf("vxedit -fr rm
%s\n", volname);
    printf("vxassist -g
rootdg -U gen make %s %lldK ",
volname,
perfilesize/1024);
    if (nstripe > 1)
        printf("\n\n
layout=stripe nstripe=%d stripe_width=%dK ",
nstripe,
STRIPE_WIDTH/1024);
    printf("user=oracle
group=dba %s\n", volstring);
}
else
{
    printf("vxassist -g
rootdg -U gen -o slow=1,iosize=2048 "
"mirror %s ",
volname);
    if (nstripe > 1)
        printf("\n\n
layout=stripe nstripe=%d stripe_width=%dK ",
nstripe,
STRIPE_WIDTH/1024);
    printf("%s\n",
volstring);
}
if (m || !mirror)
    printf("&\n");
}
}
printf("wait\n");
printf("echo Done with %s at `date`\n", name);
}

-----
sae1.vx61.disks
-----
1. c1t0d0 <ZITEL
1. c1t1d0 <ZITEL
1. c1t2d0 <ZITEL
1. c1t3d0 <ZITEL
1. c1t4d0 <ZITEL
1. c1t5d0 <ZITEL
1. c1t6d0 <ZITEL
1. c2t0d0 <ZITEL
1. c2t1d0 <ZITEL
1. c2t2d0 <ZITEL
1. c2t3d0 <ZITEL
1. c2t4d0 <ZITEL
1. c2t5d0 <ZITEL
1. c2t6d0 <ZITEL
1. c3t0d0 <ZITEL
1. c3t1d0 <ZITEL
1. c3t2d0 <ZITEL
1. c3t3d0 <ZITEL
1. c3t4d0 <ZITEL
1. c3t5d0 <ZITEL
1. c3t6d0 <ZITEL
1. c4t0d0 <ZITEL
1. c4t1d0 <ZITEL
1. c4t2d0 <ZITEL
1. c4t3d0 <ZITEL
1. c4t4d0 <ZITEL
1. c4t5d0 <ZITEL
1. c4t6d0 <ZITEL
1. c5t0d0 <ZITEL
1. c5t1d0 <ZITEL
1. c5t2d0 <ZITEL
1. c5t3d0 <ZITEL
1. c5t4d0 <ZITEL
1. c5t5d0 <ZITEL
1. c5t6d0 <ZITEL
1. c6t0d0 <ZITEL
1. c6t1d0 <ZITEL
1. c6t2d0 <ZITEL
1. c6t3d0 <ZITEL
1. c6t4d0 <ZITEL
1. c6t5d0 <ZITEL
1. c6t6d0 <ZITEL
1. c7t0d0 <ZITEL
1. c7t1d0 <ZITEL
1. c7t2d0 <ZITEL
1. c7t3d0 <ZITEL
1. c7t4d0 <ZITEL
1. c7t5d0 <ZITEL
1. c7t6d0 <ZITEL
1. c8t0d0 <ZITEL
1. c8t1d0 <ZITEL
1. c8t2d0 <ZITEL
1. c8t3d0 <ZITEL
1. c8t4d0 <ZITEL
1. c8t5d0 <ZITEL
1. c8t6d0 <ZITEL
1. c9t0d0 <ZITEL
1. c9t1d0 <ZITEL
1. c9t2d0 <ZITEL
1. c9t3d0 <ZITEL
1. c9t4d0 <ZITEL
1. c9t5d0 <ZITEL
1. c9t6d0 <ZITEL
1. c10t0d0 <ZITEL
1. c10t1d0 <ZITEL
1. c10t2d0 <ZITEL
1. c10t3d0 <ZITEL
1. c10t4d0 <ZITEL
1. c10t5d0 <ZITEL
1. c10t6d0 <ZITEL
1. c11t0d0 <ZITEL
1. c11t1d0 <ZITEL
1. c11t2d0 <ZITEL
1. c11t3d0 <ZITEL
1. c11t4d0 <ZITEL
1. c11t5d0 <ZITEL
1. c11t6d0 <ZITEL
1. c12t0d0 <ZITEL
1. c12t1d0 <ZITEL
1. c12t2d0 <ZITEL
1. c12t3d0 <ZITEL
1. c12t4d0 <ZITEL
1. c12t5d0 <ZITEL
1. c12t6d0 <ZITEL
1. c13t0d0 <ZITEL
1. c13t1d0 <ZITEL
1. c13t2d0 <ZITEL
1. c13t3d0 <ZITEL
1. c13t4d0 <ZITEL
1. c13t5d0 <ZITEL
1. c13t6d0 <ZITEL
1. c14t0d0 <ZITEL
1. c14t1d0 <ZITEL
1. c14t2d0 <ZITEL
1. c14t3d0 <ZITEL
1. c14t4d0 <ZITEL
1. c14t5d0 <ZITEL
1. c14t6d0 <ZITEL
1. c15t0d0 <ZITEL
1. c15t1d0 <ZITEL
1. c15t2d0 <ZITEL
1. c15t3d0 <ZITEL
1. c15t4d0 <ZITEL
1. c15t5d0 <ZITEL
1. c15t6d0 <ZITEL
1. c16t0d0 <ZITEL

```

1. c16t1d0 <ZITEL
1. c16t2d0 <ZITEL
1. c16t3d0 <ZITEL
1. c16t4d0 <ZITEL
1. c16t5d0 <ZITEL
1. c16t6d0 <ZITEL
1. c17t0d0 <ZITEL
1. c17t1d0 <ZITEL
1. c17t2d0 <ZITEL
1. c17t3d0 <ZITEL
1. c17t4d0 <ZITEL
1. c17t5d0 <ZITEL
1. c17t6d0 <ZITEL
1. c18t0d0 <ZITEL
1. c18t1d0 <ZITEL
1. c18t2d0 <ZITEL
1. c18t3d0 <ZITEL
1. c18t4d0 <ZITEL
1. c18t5d0 <ZITEL
1. c18t6d0 <ZITEL
1. c19t0d0 <ZITEL
1. c19t1d0 <ZITEL
1. c19t2d0 <ZITEL
1. c19t3d0 <ZITEL
1. c19t4d0 <ZITEL
1. c19t5d0 <ZITEL
1. c19t6d0 <ZITEL
1. c20t0d0 <ZITEL
1. c20t1d0 <ZITEL
1. c20t2d0 <ZITEL
1. c20t3d0 <ZITEL
1. c20t4d0 <ZITEL
1. c20t5d0 <ZITEL
1. c20t6d0 <ZITEL
1. c21t0d0 <ZITEL
1. c21t1d0 <ZITEL
1. c21t2d0 <ZITEL
1. c21t3d0 <ZITEL
1. c21t4d0 <ZITEL
1. c21t5d0 <ZITEL
1. c21t6d0 <ZITEL
1. c22t0d0 <ZITEL
1. c22t1d0 <ZITEL
1. c22t2d0 <ZITEL
1. c22t3d0 <ZITEL
1. c22t4d0 <ZITEL
1. c22t5d0 <ZITEL
1. c22t6d0 <ZITEL
1. c23t0d0 <ZITEL
1. c23t1d0 <ZITEL
1. c23t2d0 <ZITEL
1. c23t3d0 <ZITEL
1. c23t4d0 <ZITEL
1. c23t5d0 <ZITEL
1. c23t6d0 <ZITEL
1. c24t0d0 <ZITEL
1. c24t1d0 <ZITEL
1. c24t2d0 <ZITEL
1. c24t3d0 <ZITEL
1. c24t4d0 <ZITEL
1. c24t5d0 <ZITEL
1. c24t6d0 <ZITEL
1. c25t0d0 <ZITEL
1. c25t1d0 <ZITEL
1. c25t2d0 <ZITEL
1. c25t3d0 <ZITEL
1. c25t4d0 <ZITEL
1. c25t5d0 <ZITEL
1. c25t6d0 <ZITEL
1. c26t0d0 <ZITEL
1. c26t1d0 <ZITEL
1. c26t2d0 <ZITEL
1. c26t3d0 <ZITEL
1. c26t4d0 <ZITEL
1. c26t5d0 <ZITEL
1. c26t6d0 <ZITEL
1. c27t0d0 <ZITEL
1. c27t1d0 <ZITEL
1. c27t2d0 <ZITEL
1. c27t3d0 <ZITEL
1. c27t4d0 <ZITEL
1. c27t5d0 <ZITEL
1. c27t6d0 <ZITEL

1. c28t0d0 <ZITEL
1. c28t1d0 <ZITEL
1. c28t2d0 <ZITEL
1. c28t3d0 <ZITEL
1. c28t4d0 <ZITEL
1. c28t5d0 <ZITEL
1. c28t6d0 <ZITEL
1. c29t0d0 <ZITEL
1. c29t1d0 <ZITEL
1. c29t2d0 <ZITEL
1. c29t3d0 <ZITEL
1. c29t4d0 <ZITEL
1. c29t5d0 <ZITEL
1. c29t6d0 <ZITEL
1. c30t0d0 <ZITEL
1. c30t1d0 <ZITEL
1. c30t2d0 <ZITEL
1. c30t3d0 <ZITEL
1. c30t4d0 <ZITEL
1. c30t5d0 <ZITEL
1. c30t6d0 <ZITEL
1. c31t0d0 <ZITEL
1. c31t1d0 <ZITEL
1. c31t2d0 <ZITEL
1. c31t3d0 <ZITEL
1. c31t4d0 <ZITEL
1. c31t5d0 <ZITEL
1. c31t6d0 <ZITEL
1. c32t0d0 <ZITEL
1. c32t1d0 <ZITEL
1. c32t2d0 <ZITEL
1. c32t3d0 <ZITEL
1. c32t4d0 <ZITEL
1. c32t5d0 <ZITEL
1. c32t6d0 <ZITEL
1. c33t0d0 <ZITEL
1. c33t1d0 <ZITEL
1. c33t2d0 <ZITEL
1. c33t3d0 <ZITEL
1. c33t4d0 <ZITEL
1. c33t5d0 <ZITEL
1. c33t6d0 <ZITEL
1. c34t0d0 <ZITEL
1. c34t1d0 <ZITEL
1. c34t2d0 <ZITEL
1. c34t3d0 <ZITEL
1. c34t4d0 <ZITEL
1. c34t5d0 <ZITEL
1. c34t6d0 <ZITEL
1. c35t0d0 <ZITEL
1. c35t1d0 <ZITEL
1. c35t2d0 <ZITEL
1. c35t3d0 <ZITEL
1. c35t4d0 <ZITEL
1. c35t5d0 <ZITEL
1. c35t6d0 <ZITEL
1. c36t0d0 <ZITEL
1. c36t1d0 <ZITEL
1. c36t2d0 <ZITEL
1. c36t3d0 <ZITEL
1. c36t4d0 <ZITEL
1. c36t5d0 <ZITEL
1. c36t6d0 <ZITEL
1. c37t0d0 <ZITEL
1. c37t1d0 <ZITEL
1. c37t2d0 <ZITEL
1. c37t3d0 <ZITEL
1. c37t4d0 <ZITEL
1. c37t5d0 <ZITEL
1. c37t6d0 <ZITEL
1. c38t0d0 <ZITEL
1. c38t1d0 <ZITEL
1. c38t2d0 <ZITEL
1. c38t3d0 <ZITEL
1. c38t4d0 <ZITEL
1. c38t5d0 <ZITEL
1. c38t6d0 <ZITEL
1. c39t0d0 <ZITEL
1. c39t1d0 <ZITEL
1. c39t2d0 <ZITEL
1. c39t3d0 <ZITEL
1. c39t4d0 <ZITEL
1. c39t5d0 <ZITEL

1. c39t6d0 <ZITEL
1. c40t0d0 <ZITEL
1. c40t1d0 <ZITEL
1. c40t2d0 <ZITEL
1. c40t3d0 <ZITEL
1. c40t4d0 <ZITEL
1. c40t5d0 <ZITEL
1. c40t6d0 <ZITEL
1. c41t0d0 <ZITEL
1. c41t1d0 <ZITEL
1. c41t2d0 <ZITEL
1. c41t3d0 <ZITEL
1. c41t4d0 <ZITEL
1. c41t5d0 <ZITEL
1. c41t6d0 <ZITEL
1. c42t0d0 <ZITEL
1. c42t1d0 <ZITEL
1. c42t2d0 <ZITEL
1. c42t3d0 <ZITEL
1. c42t4d0 <ZITEL
1. c42t5d0 <ZITEL
1. c42t6d0 <ZITEL
1. c43t0d0 <ZITEL
1. c43t1d0 <ZITEL
1. c43t2d0 <ZITEL
1. c43t3d0 <ZITEL
1. c43t4d0 <ZITEL
1. c43t5d0 <ZITEL
1. c43t6d0 <ZITEL
1. c44t0d0 <ZITEL
1. c44t1d0 <ZITEL
1. c44t2d0 <ZITEL
1. c44t3d0 <ZITEL
1. c44t4d0 <ZITEL
1. c44t5d0 <ZITEL
1. c44t6d0 <ZITEL
1. c45t0d0 <ZITEL
1. c45t1d0 <ZITEL
1. c45t2d0 <ZITEL
1. c45t3d0 <ZITEL
1. c45t4d0 <ZITEL
1. c45t5d0 <ZITEL
1. c45t6d0 <ZITEL
1. c46t0d0 <ZITEL
1. c46t1d0 <ZITEL
1. c46t2d0 <ZITEL
1. c46t3d0 <ZITEL
1. c46t4d0 <ZITEL
1. c46t5d0 <ZITEL
1. c46t6d0 <ZITEL
1. c48t0d0 <ZITEL
1. c48t1d0 <ZITEL
1. c48t2d0 <ZITEL
1. c48t3d0 <ZITEL
1. c48t4d0 <ZITEL
1. c48t5d0 <ZITEL
1. c48t6d0 <ZITEL
1. c49t0d0 <ZITEL
1. c49t1d0 <ZITEL
1. c49t2d0 <ZITEL
1. c49t3d0 <ZITEL
1. c49t4d0 <ZITEL
1. c49t5d0 <ZITEL
1. c49t6d0 <ZITEL
1. c50t0d0 <ZITEL
1. c50t1d0 <ZITEL
1. c50t2d0 <ZITEL
1. c50t3d0 <ZITEL
1. c50t4d0 <ZITEL
1. c50t5d0 <ZITEL
1. c50t6d0 <ZITEL
1. c51t0d0 <ZITEL
1. c51t1d0 <ZITEL
1. c51t2d0 <ZITEL
1. c51t3d0 <ZITEL
1. c51t4d0 <ZITEL
1. c51t5d0 <ZITEL
1. c51t6d0 <ZITEL
1. c52t0d0 <ZITEL
1. c52t1d0 <ZITEL
1. c52t2d0 <ZITEL
1. c52t3d0 <ZITEL
1. c52t4d0 <ZITEL

1. c52t5d0 <ZITEL
1. c52t6d0 <ZITEL
1. c53t0d0 <ZITEL
1. c53t1d0 <ZITEL
1. c53t2d0 <ZITEL
1. c53t3d0 <ZITEL
1. c53t4d0 <ZITEL
1. c53t5d0 <ZITEL
1. c53t6d0 <ZITEL
1. c54t0d0 <ZITEL
1. c54t1d0 <ZITEL
1. c54t2d0 <ZITEL
1. c54t3d0 <ZITEL
1. c54t4d0 <ZITEL
1. c54t5d0 <ZITEL
1. c54t6d0 <ZITEL
1. c55t0d0 <ZITEL
1. c55t1d0 <ZITEL
1. c55t2d0 <ZITEL
1. c55t3d0 <ZITEL
1. c55t4d0 <ZITEL
1. c55t5d0 <ZITEL
1. c55t6d0 <ZITEL
1. c56t0d0 <ZITEL
1. c56t1d0 <ZITEL
1. c56t2d0 <ZITEL
1. c56t3d0 <ZITEL
1. c56t4d0 <ZITEL
1. c56t5d0 <ZITEL
1. c56t6d0 <ZITEL
1. c57t0d0 <ZITEL
1. c57t1d0 <ZITEL
1. c57t2d0 <ZITEL
1. c57t3d0 <ZITEL
1. c57t4d0 <ZITEL
1. c57t5d0 <ZITEL
1. c57t6d0 <ZITEL
1. c58t0d0 <ZITEL
1. c58t1d0 <ZITEL
1. c58t2d0 <ZITEL
1. c58t3d0 <ZITEL
1. c58t4d0 <ZITEL
1. c58t5d0 <ZITEL
1. c58t6d0 <ZITEL
1. c59t0d0 <ZITEL
1. c59t1d0 <ZITEL
1. c59t2d0 <ZITEL
1. c59t3d0 <ZITEL
1. c59t4d0 <ZITEL
1. c59t5d0 <ZITEL
1. c59t6d0 <ZITEL
1. c60t0d0 <ZITEL
1. c60t1d0 <ZITEL
1. c60t2d0 <ZITEL
1. c60t3d0 <ZITEL
1. c60t4d0 <ZITEL
1. c60t5d0 <ZITEL
1. c60t6d0 <ZITEL
1. c61t0d0 <ZITEL
1. c61t1d0 <ZITEL
1. c61t2d0 <ZITEL
1. c61t3d0 <ZITEL
1. c61t4d0 <ZITEL
1. c61t5d0 <ZITEL
1. c61t6d0 <ZITEL

```

-----
From df -k
-----
Filesystem          kbytes    used    avail capacity
Mounted on
/dev/dsk/c0t0d0s0   290065    30863  230202    12%    /
/dev/dsk/c0t0d0s6   290065   191504   69561    74%    /usr
/proc                0         0         0         0%    /proc
fd                   0         0         0         0%
/dev/fd
/dev/dsk/c0t0d0s3    771110   12958   681042     2%    /var
/dev/dsk/c0t0d0s7   5033718  4749730  82640    99%
/export
/dev/dsk/c0t0d0s5    482824   95956   338588    23%    /opt
swap                 23357896      504  23357392     1%
/tmp
/dev/dsk/c0t1d0s0    290065    28798  232267    12%    /mnt

```

goby37:/export/GOODies 2262576 1758592 277736 87%
 /opt/GOODies
 goby41:/goby9/djm 5780456 3547712 2059336 64%
 /home/djm
 Sat Mar 8 09:37:29 PST 1997

iOMxk3j5RCROR1 5BOBMn2113hJOS25ij7i23C2
 NykmCOg16P1RL61RPP4mPwn13R7Q5BjOSMR3

23-APR-94 502436 40750060 5-LOW 0
 Clerk#000289212 F
 275526.94
 wO56kB3Cj 6 LyNC5 5kwCMSQS

29-MAR-92 5024378276846 2-HIGH 0 Clerk#000172567 F
 110375.31
 3PymR2MPAomk6MN6SNSh2n160i0jLAAyM 75 OzzAi07M21x7An2B
 Mj0AQXA

26-JUL-98 502438 13948150 5-LOW 0
 Clerk#000232748 O
 233757.8

l3hMR2ymPklmgij6wMLL50NLBj45gzCSOLBR20jB5j3iNkMR0MhPs4hjNB
 OMASg-KAl

06-MAY-94 502439 21543269 3-MEDIUM 0
 Clerk#000094311 F
 324140.25
 MlQk3lR1M0w0nQ34ni6hzPjlny2z

05-SEP-97 5024647714058 3-MEDIUM 0 Clerk#000175440
 38036.57
 0lh Sy3Qj3LxAyM1 yAmxCNLk7 4Q56L52 hLzlh3x3R5Pz6RB50M2zB

25-APR-98 5024655167234 5-LOW 0 Clerk#000212253 O
 188889.19
 gOlzMOri0hQmkAh563mkwB0PSA6lSjQA4wkMLlRyNyy051xAghBA 1PM

07-SEP-97 502466 40694284 5-LOW 0
 Clerk#000029715 O
 85924.38
 CB0Bz2vhwS6CSnh7R y mLmi 7M2x 3N6mhzngPhjO z7nL zLjQ1y361P
 RvLgBmMLNOLA4

14-APR-96 502467 38828047 2-HIGH 0
 Clerk#000203811 O
 69614.56
 kQA67LnC4j41NxBO 730Ozi wlz7kBMng37y1xxS2MCQx ixm

19-NOV-92 502468 37980901 5-LOW 0
 Clerk#000049911 F
 152097
 nlzjjLP1h1klBNLQ1NR5GjN3y5Rx Mly5

10 rows selected.

partsupp

PS_PARTKEY PS_SUPPKEY PS_SUPPLYCOST PS_AVAILQTY

 PS_COMMENT

24997630 997631 544.48 8307
 1B5g
 BWR5QzMAz0ixNkN2CPSxN5OgSgLAMkkBmxniwO36xwLz1z21RjA4x3360y2

24997630 1747639 456.35 1979
 wlp6iL3jLlhm9NwAQ2P
 Bli0LynS4mkC37QKA4Z2SQB7L3nR06y71BO5w27L5RmO lwiSM4SwNmgm
 RlNmhw1Q72PO73 MORSC4kh

24997630 2497647 658.18 8889

YlmcS14CON1MwAQ57zS10jwLwSg3Snpi45ON
 0Rgm41zjkmC2g3I644S1i107ziizEBNBAGKOK3I
 4MicM4xC206kh7xkx31ijmzhPzP473S7gLn6nn7mj7zwwx

24997630 247655 206.69 8657

lh0Lkw6m4g2l7mRyBN3Sx5Nzi_g5BniqQBR0hk1x
 MlgImAmh3R7mknLijjLRnN325_6nmGjInOSQP

OhB_17vnmjnnNS4i7
 PR2Cgy60103mkfign7jOj7Mi0wMA0R0LQw50A4L3P
 CNMzZ15N4ARjQLP6K
 wixiCPmA

24997631 997632 551.72 2411

mCi05xBL5AchB058_6LLC5
 KRLACX76mSgZB0n4370ySACQSBciOmRmOximlykiPP2gh5liNQxOyOmwh
 NG01x2NzxPgRikhy74i0lk3_A7M6g
 Wmyxm6NwLZR1003AJkL2kgCNBNBsgSL531nSiS2QqhkPmh66i
 B7BM BkONkgj 0Mgy63iRRQ1i11

24997631 1747640 280.79 9266

3kShx1zMNh7yBh2SBwPS6wQ2g3A2mSh6B1NPQN37jAy3iNmM3PwB3lk2O
 BxOCakZj6LjSk7Rw2Q3

24997631 2497648 163.6 3646

Appendix G. Initial Ten Rows of Tables

lineitem

L_SHIPDATE L_ORDERKEY L_DISCOUNT L_EXTENDEDPRICE L_SUPPKEY
 L_QUANTITY L

L_PARTKEY L L_TAX L_COMMITD L_RECEIPT L_SHIPMODE
 L_LINENUMBER

L_SHIPINSTRUCT L_COMMENT

24-APR-94 945411 0 59523.18 2236364

21736349 F 0 18-FEB-94 27-APR-94 TRUCK 3
 TAKE BACK RETURN 3Rh071BP2OCNwR2CCx1MxhmKxj

12-MAR-94 945411 .02 79977.55 1559809

42809794 F .07 16-MAR-94 27-MAR-94 MAIL 4
 NONE yCjRBwzkC5QQyOMk4igACBSh75wN5OxjLxyl31ykCz

23-MAR-94 945411 .1 35030.38 2945514

44195499 F .06 17-FEB-94 23-APR-94 FOB 5
 NONE Bjm6CMxL3PwyClP zwzhzyzhC zNxzN

31-AUG-98 945412 .04 26069.96 1242857

53742822 O 0 06-JUL-98 04-SEP-98 REG AIR 1
 DELIVER IN PERSON jy0AzPMY0MnAk6002ylyiR7yinkOj wOjy4x

04-JUN-98 945412 .02 48781.5 2105541 34 N

15605530 O .02 12-JUN-98 19-JUN-98 SHIP 2
 TAKE BACK RETURN yzNoj3w5xQk

24-AUG-98 945412 .1 10494.18 389150

22139128 O .08 23-JUN-98 12-SEP-98 TRUCK 3
 COLLECT COD 5knCOBgnjjLn51

10-AUG-98 945412 .01 1203.58 63203 1 N

32313192 O 0 03-AUG-98 13-AUG-98 AIR 4
 DELIVER IN PERSON 1y60iCj0462wg

12-MAY-98 945412 .04 86571.5 1561831 50 N

27811821 O .03 14-JUL-98 18-MAY-98 RAIL 5
 TAKE BACK RETURN lRMNBNS0xN53C0z5L1kn231BRw4i0AniQ

30-JUL-98 945412 .07 37793.52 1773451

33273428 O .01 07-AUG-98 26-AUG-98 SHIP 6
 COLLECT COD yPriXlmhjng24

09-JUN-96 945413 .06 8076.81 1749181 7 N

26499156 O .04 05-JUL-96 10-JUN-96 TRUCK 1
 TAKE BACK RETURN Ohm54yN72BiRgh52RNLAm60LA QAS1CMSR3xygS

10 rows selected.

orders

O_ORDERDATE O_ORDERKEY O_CUSTKEY O_ORDERPRIORITY
 O_SHIPRIORITY O_CLERK O

O_TOTALPRICE

O_COMMENT

11-JUL-95 502435 11793941 4-NOT SPECIFIED 0

Clerk#000098173 O
 92934.36

LixN5F0 iCzvm01mB0n6zOmLS MJA00kAM7LkA6A5n5
Q6B2x23xIR1015MgiBwMC6L10kz 32yyw1n
jz00gyLwB30Mh5CQ6S

24997631 247656 689.62 2435
4jMAkw01Q2Ans OM5C0227CCBxS
7hmzYk9591wF0hm1IQLS2Rim120S7Lk2Qj120nnzym2h3iyk0m
411yjAzC6gACzm

24997632 997633 256.97 5307
5hi kxjSAA4RNSCnO2R5RAO6134Rmi34OLALB0jN1ww4m0xRNMSxj2xgzkC

24997632 1747641 436.75 1719
5lvsyZAOnlgLN44Q362wMBS6Qmjm3714yzL7kB6kLOA
94N0K00LGNB7z9PClyRIARPOQySIA15NL
14mlv6cm71 NgXR176Lyz2nzycN0L6B06gw1Bn71LAjN405nz
h1y1zjQBjQiy6KO2yA61zma32g156
6350MBN

10 rows selected.

parts

P_PARTKEY P_TYPE P_SIZE P_BRAND

P_NAMEP_CONTAINE

P_MFGR P_RETAILPRICE P_COMMENT

12889619 LARGE BURNISHED STEEL 31 Brand#12
dim ghost almond ivory maroonJUMBO JAR
Manufacturer#11607.97 M103P14zL24L4gk7mSwsL

12889620 MEDIUM BRUSHED TIN 43 Brand#31
maroon medium dodger purple forestWRAP CASE
Manufacturer#31608.98 Qlnknh3QyNBO27B2gAiP4

12889621 STANDARD ANODIZED COPPER 37 Brand#42
olive blanched grey powder rose SM CASE
Manufacturer#41609.98 7 mLhkhR6R5l iQNCBjkr

12889622 MEDIUM BURNISHED STEEL 12 Brand#32
plum khaki slate lavender blush SM BOX
Manufacturer#31610.98 zwCOLQR3P21

12889623 LARGE POLISHED BRASS 3 Brand#21
beige antique puff forest gainsboroWRAP CAN
Manufacturer#21611.98 nmR4C010lnk0m

12889624 PROMO BURNISHED TIN 25 Brand#12
goldenrod spring snow lavender navajoLG BOX
Manufacturer#11612.98 y4yzxhR10N1jBg3j120

12889625 MEDIUM PLATED COPPER 13 Brand#21
blanched tomato bisque burlywood lavenderJUMBO JAR
Manufacturer#21613.98 xRk1OLM gSin2y

12889626 STANDARD PLATED COPPER 7 Brand#31
mint chiffon cyan thistle greyWRAP CASE
Manufacturer#31614.98 R2MSykPMAO2kB

12889627 LARGE ANODIZED COPPER 5 Brand#45
coral cornsilk moccasin navy royallG PACK
Manufacturer#41615.98 z317kA

12889628 LARGE BRUSHED COPPER 2 Brand#23
drab lemon turquoise smoke mistyJUMBO CASE
Manufacturer#21616.98 1hMwQn0zxAIML5B01 mjR

10 rows selected.

customer

C_CUSTKEY C_MKTSEGME C_NATIONKEY C_NAME

C_ADDRESS C_PHONE C_ACCTBAL

C_COMMENT

6788402 HOUSEHOLD 11 Customer#006788402
C1SmAPCx3AQ6 21-775-834-2631 2418.33

m10h66NjQ7B7kSLncXik1Pz4jyt6w
C1xh2Mlxj5n0Cz1y6Q0gjnSL1140Q1lnS05 64z5P 40iB01
mMMA4i

6788403 BUILDING 13 Customer#006788403
jLxOMLQyiz2B0wwRzL 23-243-343-5786 8869.09
6z415g1gl10Q mwCk1ixSgxhL1Lyzxii5w23MLA i70S1Ngzn310

6788404 FURNITURE 17 Customer#006788404
kS6gmglQCQ3AM17k7xjgCOQ133nL5B 27-201-586-2350 -154.37
wgmNk7ABQBAvg5i3BNijmMxmMgLOQ3Ch25
QM3Bh1RyL500L6Aw568jBhmR0WlKX1 SA212

6788405 BUILDING 17 Customer#006788405
3Q5w2wAx7703k5 27-148-343-2227 -553.1
Rgjl mgxmCjLlG353076j4PzBwnCnPxxx4BOzkwBMOi 2wj6PmlkmM 0

6788406 FURNITURE 16 Customer#006788406
73wwlxhLLRS7B5N75CSgA0i513 26-622-458-2828 7923.22
iz4AN1AA36wOx2g6LLwAChOAw iA121Bk1Mw Pyh6A02RgWCO1

6788407 AUTOMOBILE3 Customer#006788407
y1 MkLBB41i4 RAQw67ngQh 13-515-634-5986 8693.18
2Cn6xQR1y7L gziR5QBmnw6LjCygx2kNznhw7znN65C7R1im31CLzQzn1

6788408 FURNITURE 14 Customer#006788408
5iRh1720B 24-607-426-2443 2412.7
QzAip17iw57RN1SSi5LBR7mill1C151M0jBk7LCSALNxlz1SByAn7xgP
S2RR0XgzPw0024yNKMBY61S
ykh1151L4QNO

6788409 AUTOMOBILE 20 Customer#006788409
LhRgjM0B1A0S1Sn2kjNjMR 6PlAxLynSxL2gO62 30-646-190-
12697 2143.86
5y7gilB2yOlPvLw54hSgBxN6C1ln4R 12715A3 SLCmwi05L17LCLwy0
x6nnwwLCS271K41nAw71n

6788410 AUTOMOBILE 13 Customer#006788410
kQjnAw0g6Q2S3S03j54hg 23-872-275-1818 4746.73
MOh1M1Lj5hyOQNxBhN14BkmN2L6hM4SCLkzjzNONR1CSR

6788411 AUTOMOBILE 24 Customer#006788411
A4jhnPBpN6MghN1jP1 34-388-353-3945 4304.14
wvdkRgww8y
2chnwYcJr3zy2wgxPnO0j5hxA7h1NSyORykIC7n01Lh7R510Qx0mjPRI

10 rows selected.

supplier

S_SUPPKEY S_NATIONKEY

S_COMMENT

S_NAME S_ADDRESS

S_PHONE S_ACCTBAL

2057700 17
kQhgQwBihhwSh 3MN7mN6ji51NMzQS iMkw4LOPB3ShNy 16wBkxLj4
5A3mm4AQBNzyCxQBNI30B41x
znklg5nN70
Supplier#002057700 7ynOgkOPwxNSjg1jOjSck3n2jkyg4R4z
27-851-274-8878 -338.05

2057701 19
iw7LQP35L7j63Mn32SNC3AilwCQ17C1y0
Supplier#002057701 nSBPCrn3S01xm37ha04i7C4mjnLzyyQ
29-659-629-5606 3282.07

2057702 12
623m4xOSBRRALxnjNBA6MNHm1hNn5NmChxjCazM56jzA5AwzgrN0Mhk7
5NyL03PO1zBRkni123L0511
BANhnh5z
Supplier#002057702 M05S50izQCSh MAA6 BOQB6xyOAhCj xjPjBnn
22-987-392-4230 5723.99

2057703 0
PwNagQ3LkPw6CzmClOmSij kNM3Nx 5y5S4B5zBmNBmm06SnSjgz
jCO0yz4Q3Lh7j BPO 10wgmP4n
Supplier#002057703 ih7MAj5gNi4Mj2gAPg50BmnML506 zhxk3A45mn
10-694-906-6377 6057.79

2057704 14
R32lmB1n5NkO2ykm426g L 1 iicKjw
NigBj5wrjPRRwZNO7n0R57Iz5LwS5

Supplier#002057704 jLjRQly6RONS3P5xMMNL5SiySSQmP
 24-734-237-9875 6377.79

2057705 9
 mOm16iA11NSBLCzi0y2Smh5hN66 PP1LhkgN 3yQjMkBjCjwB3Cnkz1
 6y5PCB
 Supplier#002057705 mlNyM0 ALPN5k1Cn77
 19-250-580-7923 5480.86

2057706 6
 An51i5l3SwkOR36M0n 6jPS 7iPCR2hPOix7zyAyM7P
 Supplier#002057706 P5QMSxRO5Sx2m1AmAggAgCkwB1gyB
 16-894-652-8197 770.46

2057707 22
 5y41CmhBBSL37AM6iSMQy27i7C7BBhiaQkzikMOxii2k7y
 Supplier#002057707 NjClRy01k7 lyCL
 32-472-362-8297 29.52

2057708 3
 BhBlwLh24m6znjRLAOAg0 iwAMSR3LCAMkBRkg00zjnlx023giBhkAP
 Supplier#002057708 i6xiwz50yyk304hzC2kBO1Rkw
 13-639-907-6883 -971.54

2057709 10
 5iNNiSl C45 x NCRA3AC O16xB3hSax BO1 RowS67QM
 Supplier#002057709 50xQP14NA1Pk
 20-818-732-6605 4311.56

xSx31zz31Cl1z40Anmm05AjiOx3C3AMNOgC0kACgwnng3glP7LLLywlQy7R

1 AMERICA
 kqy31SpC72k6zLz0LP3k2L4QB1QL106730j01SPj0ngQ7CO100SBgmgRQ
 4jgPCMK2IA4251klyAR4y
 BRAR4Cm5miNw 4j113mMnxw17B

2 ASIA
 NSq6x1M1A11zm6mQR0Aix nhRA77NgRxBwL1M6PY
 RjySB3RLWkyPKwMM2R1BQ xAzkOgkjm110gAgni
 nP5inmNmR76M1ijMS3S2zxONR15

3 EUROPE
 z1SL7Owg12hMBL51h1z0M45QkjShwSyi004MLOh7wn1ARLQPyPayAii1576
 11L17A1hR1S RQ4SLny7B
 2Ryj5P66MLhn NxhwB4C3ig0S0

4 MIDDLE EAST
 R1lxmhPLz3Cy2mNlg4QMBnNASM ACki MPki70i

Appendix H. Hardware Pricing: Reseller Price Quotes

10 rows selected.

nation

N_NATIONKEY	N_NAME	N_REGIONKEY
0	ALGERIA0	
2	Cxhl7 Lliwk6hMh300izngN32CPwCikyLk6khMzSRA	
1	ARGENTINA1	
z	Qn3Okwz1Ln7PLS3OhCgn56kP5PyRikgi1B7LL	
2	BRAZIL1	
g	lms0nACAmRCi2k1ki7RCpNgPxnCOjNg4k OiAg57COSom1NwCnOyLx40R SC y20gPPAKNR5hxRhr5 OmgS1iPQQzNaxPL30n67OgyC 1617Sh4LS	
3	CANADA1	
4	yMO AhnQ5Lh wzQAM662Aw1ByCl7CxmzRwNR5nA1O4 x	
4	EGYPT4	
1	lim5126 Cxj NMQmLxOikni02j2m3Ah4yNR1QQiL507j2QSlyN	
5	ETHIOPIA0	
NS	7n LSOP Oz5n1A1B2S02nN01Mh4SBxP iRhBO 047R26 2B1M	
6	FRANCE3	
3	mjmizl S 3L3k2hNNhN1P4w370xRxyN15wn	
7	GERMANY3	
z	nOP4RkwO CmzBB 516mAg 1Byw4OM3QyNPA	
8	INDIA2	
MN	1R5RCiRMj1111w1N7Myn M1ly1N1MmB017PL4C KkxQK9PQ713w6B67R2QKOO40x14Q21w76JRL71 1hR5Q 0xC7Rrm5iQ2Nax2LiBm3Qi027j	
9	INDONESIA2	
Sj	PmQO71Lj 7ABj6MxlAQk3nLwi73BPxzCwjzMn4zLzgg6nzz0j0w zXC66gP6yKRPMg	

10 rows selected.

region

R_REGIONKEY	R_NAME
0	AFRICA



CATtechnology Sales Quotation

Quotation Number: 032097
Quote Date: 3/20/97

To: Brad Carlile Manager Strategic Application Engineering Sun Microsystems Computer Corporation 8300 SW Creekside Place Beaverton, OR 97008 tel: 503 520-7622 fax: 503 520-7724	From: Dick Crouch CAT technology, Inc. 3031 Tisch Way San Jose, CA 95128 408-345-9199 FAX 345-9191
--	--

Ref: B. Carlile	Credit Terms: Net 30 1.5% mo past due	FOB: Org.	Page: 1 of 1
-----------------	---------------------------------------	-----------	--------------

Quote Valid for 30 days

Item	Part Number	Description	Net Price	Qty	Ext Price
Enterprise Server 10000					
1	E10000	E10000 Base Cabinet	\$142,500.00	1	\$142,500.00
2	2760A	CPU SYSTEM BOARD, UNPOP	\$45,000.00	16	\$720,000.00
3	2530A	250-MHZ ULTRASPARC MOD.	\$12,000.00	64	\$768,000.00
4	2750A	SSP	\$6,671.25	1	\$6,671.25
5	7025A	MEMORY BOARD, UNPOP.	\$7,500.00	16	\$120,000.00
6	7022A	256MB MEMORY EXPANSION	\$3,562.50	64	\$228,000.00
7	783A	ELITE 9 TRAY, 2- 9GB DISKS	\$8,175.00	1	\$8,175.00
8	6504A	RSM TRAY 7* 4.2GB DISK PER TRAY	\$12,150.00	61	\$741,150.00
9	2730A	DUAL SBUS I/O BOARD	\$5,625.00	16	\$90,000.00
10	1062A	SBUS DIFF FAST/WIDE SCSI-2	\$971.25	62	\$60,217.50
11	9681A	POWER CONTROL MODULE	\$750.00	1	\$750.00
12	9671A	FAN TRAY	\$1,500.00	16	\$24,000.00
13	3800A	POWER CORD FOR NORTH AMERICA	\$0.00	4	\$0.00
14	956A	I/O EXPANSION CABINET	\$6,000.00	6	\$36,000.00
15	979A	12 METER SCSI CABLE	\$198.75	62	\$12,322.50
16	3800A	POWER CORD FOR EXP	\$0.00	6	\$0.00
17	3875A	AC BREAKER & FILTER	\$2,250.00	4	\$9,000.00
18	9685A	48 VOLT POWER SUPPLY	\$3,000.00	8	\$24,000.00
TOTAL					\$2,990,786.25

NOTE: Above configuration are generally available and prices are guaranteed for 60days