TPC Benchmark™ H
Full Disclosure Report

Sun Microsystems Sun Fire™ 15K Server
Using Oracle 9i R2 Enterprise Edition

Submitted for Review
Report Date: April 7, 2003

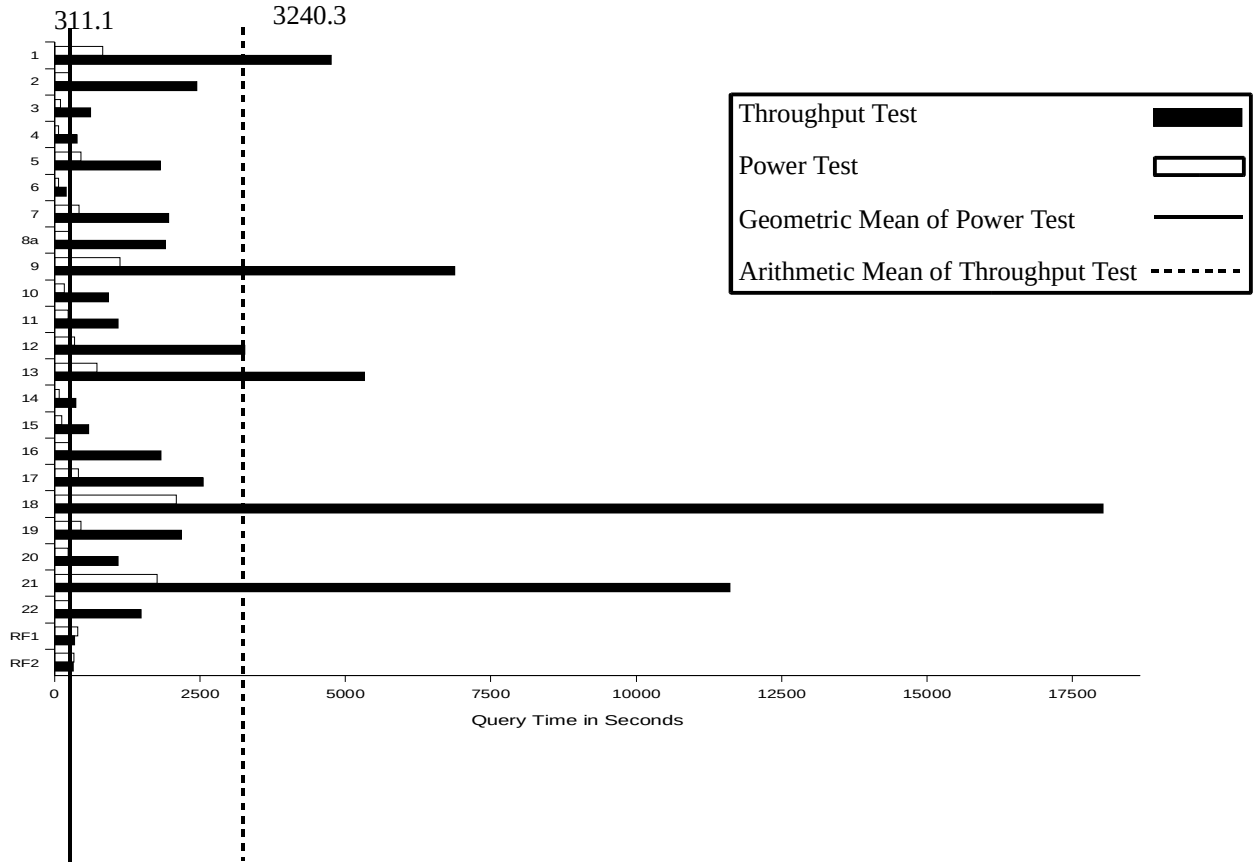
First Printing

901 San Antonio Road, Palo Alto, California 94303 U.S.A.

Sun microsystems / ORACLE	**Sun Fire™ 15K Server with Oracle9i R2**	**TPC-H Rev. 2.0** **April 7, 2003**

Total System Cost	Composite Query per Hour Metric	Price/Performance
$5,335,742	**28,948.1** QphH@3000GB	**$184** $/QphH@3000GB

Database Size	Database Manager	Operating System	Other Software	Availability Date
3000GB	Oracle 9i R2 Enterprise Edition	Solaris 9	Veritas Volume Mgr. 3.5.0	April 30, 2003



Database Load Time = 8:19	Load Includes Backup: N	Total Data Storage/Database Size=9.25
RAID (Base tables): Y	RAID (Base tables and auxiliary data structures): Y	RAID (All): N

System Configuration: Sun Fire™ 15K Server
Processors: 72 UltraSPARC™ III Cu 1200 MHz processors
Memory: 288GB memory
Disks: 33 A5200 (12x73.4GB), 2 T3 (9x36.4GB),1 S1 (2x36.4GB)
Total Storage: 27,748.2 GB (in this calculation one GB is defined as 1024*1024*1024 bytes)

Sun Fire™ 15K Server with Oracle9i R2

TPC-H Rev. 2.0

April 7, 2003

Description	Part Number	Source	Unit Price	Qty	Ext. Price	3 Yr. maint.
Server Hardware						
Sun Fire 15K Server Base	F15K-CAB3	1	706,000	1	706,000	180,144
CPU/MEM BD BNDL-4CPU@1200/16GMEM	CPUBD-482-1200	1	128,600	18	2,314,800	251,442
Opt QFE PCI card w/SW	1034A	1	1,795	1	1,795	
PCI dual Ultra-3 SCSI	X6758A	1	800	1	800	
Cable,SCSI,SCSI-3/SCSI-3, 2.0m	X1139A	1	95	1	95	
PCI I/O Assy for F15K	4575A	1	15,000	18	270,000	
Opt Pwr Cord For Enterpr.(US)	3800A	1	0	12	0	
2GB PCI Dual FC Network Adapter	X6768A	1	4,900	67	328,300	
15M Fibre Channel Cable	X9724A	1	190	134	25,460	
North American Country Kit	3508A	1	0	1	0	
Server Hardware Subtotal					3,647,250	431,586
Storage						
1601 GB StorEdge A5200 cabinet	SG-ARY563A-1601G	1	152,650	6	915,900	70,632
1606-GB Sun StorEdge A5200 array	SG-ARY571A-1606GR5		115,000	5	575,000	58,860
511-GB Sun StorEdge A5200 Array	SG-XARY570A-511G		53,000	22	1,166,000	258,984
A5000 Exp Cab Mount Kit	X9655A		250	22	5,500	
FCAL GBIC Module 100 MB/s	X6731A	1	600	66	39,600	
StorEdge S1, 2x36GB	NS-DSKS1-236GAC	1	3,195	1	3,195	2,232
655 GB StorEdge T3ES	T3BES-RK-22-655	1	88,400	1	88,400	8,460
Power Cord for StorEdge	X3858A	1	0	14	0	
Storage Subtotal					2,793,595	399,168
Server Software						
Solaris 9 Std Media	SOLZS-00AC9AYS	1	50	1	50	
SPARC Compiler C/C++	FDEIS-700-T999	1	2,995	1	2,995	3,096
Sun StorEdge Comp Mgr	SCMMS-210-R99R	1	0	1	0	
Veritas VM 3.5 for A5x00, Solaris 9	VVMGS-350-9999	1	0	1	0	
Oracle 9i Database Enterprise Edition Release 2, Named User Plus for 3 years		2	10,000	72	720,000	
Partitioning, Named User Plus for 3 years		2	2,500	72	180,000	
Oracle Database Server Support Package for 3 years		2	6,000	1		6,000
Oracle Mandatory E-Business Discount (license and support)					-181,200	
Server Software Subtotal					721,845	9,096
Volume Discounts and Support Prepayment					-2,577,076	-89,722
			Total		4,585,614	750,128
Service for all Sun products is from Sun Microsystems, Inc.			**3 Yr. Cost**		5,335,742	
Service for Oracle products is from Oracle Corp.			**QphH@3000GB**		28,948.10	
			$/QphH@3000GB		$184.32	

Notes (Source):

1. Sun Microsystems, Inc.
2. Oracle Corp. contact MaryBeth Pierantoni (see Appendix G)

Audited by: François Raab, InfoSizing, Inc. (www.sizing.com)

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

Numerical Quantities

Measurement Results:

Database Scale Factor	= 3000GB
Total Data Storage / Database Size	= 9.25
Start of database load time	= 03-26-2003 15:38:55
End of database load time	= 03-26-2003 23:57:46
Database Load Time	= 8:19
Query Streams for Throughput Test	= 8
TPC-H Power	= 34,714.4
TPC-H Throughput	= 24,139.6
TPC-H Composite Query-per-Hour Rating (QphH@3000GB)	= 28,948.1
Total System Price Over 3 Years	= $5,335,742
TPC-H Price/Performance Metric ($/QphH@3000GB)	= $184

Measurement Intervals:

Measurement Interval in Throughput Test (Ts)	= 78,742 seconds

Duration of Stream Execution:

Stream ID	Seed	Start Date	Start Time	End Date	End Time	Duration
Stream 00	326235746	3/28/03	13:20:43	3/28/03	16:31:54	03:11:11
Stream 01	326235747	3/28/03	16:32:13	3/29/03	11:28:16	18:56:03
Stream 02	326235748	3/28/03	16:32:13	3/29/03	12:57:12	20:24:59
Stream 03	326235749	3/28/03	16:32:13	3/29/03	11:47:13	19:15:00
Stream 04	326235750	3/28/03	16:32:13	3/29/03	12:12:25	19:40:12
Stream 05	326235751	3/28/03	16:32:13	3/29/03	12:37:33	20:05:20
Stream 06	326235752	3/28/03	16:32:13	3/29/03	12:23:39	19:51:26
Stream 07	326235753	3/28/03	16:32:13	3/29/03	12:40:19	20:08:06
Stream 08	326235754	3/28/03	16:32:13	3/29/03	12:35:57	20:03:44
Refresh		03/29/03	12:57:12	3/29/03	14:24:35	01:27:23

Sun Fire™ 15K Server with Oracle9i R2

TPC-H Rev. 2.0

April 7, 2003

TPC-H Timing Intervals (in seconds)

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8a	Q9	Q10	Q11	Q12
Stream 00	828.0	254.3	95.1	65.5	455.2	62.6	423.2	277.1	1126.4	165.5	227.9	344.3
Stream 01	4006.2	2626.4	778.6	392.7	1491.9	281.4	1840.1	1733.5	8043.6	1065.1	1107.5	2076.9
Stream 02	4807.4	2591.1	634.0	396.2	1776.8	97.1	2017.7	2063.0	8058.7	1054.8	1181.2	2938.1
Stream 03	4714.0	2010.7	520.4	433.6	2295.6	193.8	2163.2	2608.3	5229.7	966.4	1253.6	3400.3
Stream 04	4784.2	2164.8	359.4	353.8	1214.1	271.1	2044.7	1968.2	6449.9	1002.4	1066.4	3792.8
Stream 05	4662.9	2870.2	697.4	362.9	2024.2	200.6	1977.5	1711.5	6555.6	1136.4	1215.2	3958.5
Stream 06	4861.1	2748.0	789.6	431.2	1772.2	317.2	1831.7	2106.9	7413.5	380.4	1159.6	4270.2
Stream 07	5213.9	2114.6	663.4	368.8	1985.3	69.6	2022.6	1643.7	7044.6	814.0	1286.1	1220.5
Stream 08	5034.8	2383.2	483.6	356.3	1939.5	202.0	1771.4	1387.0	6193.5	951.1	499.4	4471.1
Minimum	4006.2	2010.7	359.4	353.8	1214.1	69.6	1771.4	1387.0	5229.7	380.4	499.4	1220.5
Average	4760.6	2438.6	615.8	386.9	1812.4	204.1	1958.6	1902.8	6873.6	921.3	1096.1	3266.0
Maximum	5213.9	2870.2	789.6	433.6	2295.6	317.2	2163.2	2608.3	8058.7	1136.4	1286.1	4471.1

	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	RF1	RF2
Stream 00	725.1	76.2	125.3	273.4	408.0	2090.3	458.5	231.9	1762.2	261.6	402.4	330.8
Stream 01	4930.9	255.5	505.3	1824.1	1388.1	17974.8	2218.7	887.5	11034.4	1700.3	368.9	316.5
Stream 02	5518.3	570.4	725.2	2283.7	3304.1	18512.3	3250.9	1220.9	8940.0	1557.4	328.3	299.3
Stream 03	5817.7	404.3	504.3	1814.2	1979.7	16894.7	2167.9	1266.7	11328.9	1331.8	341.9	314.3
Stream 04	5396.5	396.8	733.5	2194.9	1691.6	18167.5	1587.3	718.7	13265.2	1188.4	352.1	324.1
Stream 05	5534.0	302.3	657.6	1901.5	1892.5	17099.7	1404.5	1345.0	13301.4	1508.4	338.8	308.6
Stream 06	5544.8	353.7	739.0	1607.9	2944.9	16350.2	1570.4	1055.3	11800.3	1438.1	325.3	316.7
Stream 07	5921.8	163.6	205.5	1134.1	3558.5	20822.9	2143.1	1175.7	11400.7	1512.9	323.5	329.5
Stream 08	4003.1	477.5	580.1	1827.5	3692.8	18367.6	3076.4	1056.8	11815.1	1653.5	320.0	334.2
Minimum	4003.1	163.6	205.5	1134.1	1388.1	16350.2	1404.5	718.7	8940.0	1188.4	320.0	299.3
Average	5333.4	365.5	581.3	1823.5	2556.5	18023.7	2177.4	1090.8	11610.8	1486.4	337.4	317.9
Maximum	5921.8	570.4	739.0	2283.7	3692.8	20822.9	3250.9	1345.0	13301.4	1700.3	368.9	334.2

INFOSIZING

TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL CERTIFIED AUDITOR

Test Sponsors: Ray Glasstone
Manger, DSS Performance.
Oracle Corporation
100 Oracle Parkway
Redwood Shores, CA 94065

Brad Carlile
Director, Enterprise Benchmarking
Sun Microsystems, Inc.
3295 N.W. 211th Terrace
Hillsboro OR, 97124

April 7, 2003

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

Platform: **Sun Fire 15K Server**
Database Manager: **Oracle 9i R2 Enterprise Edition**
Operating System: **Solaris 9 Update 3**

The results were:

CPU (Speed)	Memory	Disks	**QphH@3000GB**
Sun Fire 15K Server			
72 x UltraSPARC III Cu (1200 MHz)	8MB E-Cache/cpu 288 GB Main	396 x 73.4 GB 20 x 36.4 GB	**28,948.1**

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

- The database records were defined with the proper layout and size
- The database population was generated using DBGEN
- The database was properly scaled to 3 TB and populated accordingly
- The compliance of the database auxiliary data structures was verified
- The database load time was correctly measured and reported
- The required ACID properties were verified and met

INFOSIZING

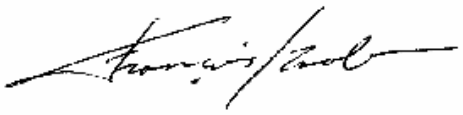
TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL
CERTIFIED AUDITOR

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

Additional Audit Notes:

The measured system included (18) 18.2 GB disks drives and (2) 9.1 GB disk drives that were substituted by (20) 36.4 GB disks, in the priced configuration. Based on the specifications of these disks and on I/O data collected during testing, it is my opinion that this substitution does not have a material effect on the reported performance.

Respectfully Yours,



François Raab
President

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

Table of Contents

TPC Benchmark H Overview

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

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

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

- Give answers to real-world business questions
- Simulate generated ad-hoc queries
- 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, Inc. and Oracle Corp. are the sponsors of this TPC-H benchmark.

1.2 Parameter Settings

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

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

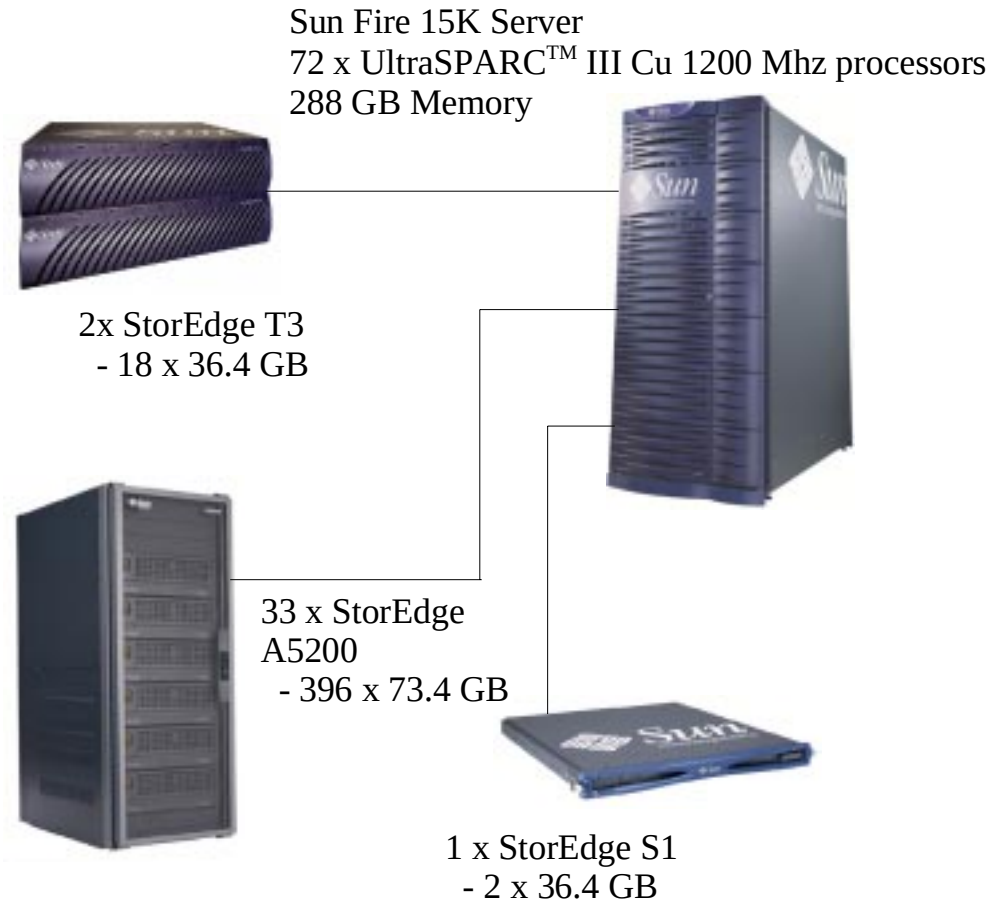
Appendix A contains the Solaris and Oracle parameters used in this benchmark.

1.3 Configuration Diagram

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

Sun Fire™ 15K Server, configured with:

- 72 UltraSPARC™ III Cu 1200 MHz processors
- 288 GB memory
- 1 Ethernet controller
- 33 A5200 disk arrays, each containing 12 x 73.4GB disk drives
  1 A5200 disk array, containing 2 x 9.1 GB disk drives (measured)
  1 S1 disk array, containing 2 x 36.4GB disk drives (priced)
- 2 T3 disk arrays, each containing 9 x 36.4GB disk drives (priced)
  9 x 18.2GB disk drives (measured)


Sun Fire 15K Server
72 x UltraSPARC™ III Cu 1200 Mhz processors
288 GB Memory
2x StorEdge T3
- 18 x 36.4 GB
33 x StorEdge
A5200
- 396 x 73.4 GB
1 x StorEdge S1
- 2 x 36.4 GB

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-H database.

2.2 Physical Organization

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

No record clustering or index clustering was used. Column ordering was changed 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.

Horizontal partitioning was used for all tables except NATION and REGION. Refer to the table/index create statements in Appendix B for more details.

2.4 Replication

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

No replication was used.

3. Clause 2 Queries and Refresh Functions

3.1 Query Language

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

SQL was the query language used to implement all queries.

3.2 Verifying Method for Random Number Generation

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

TPC supplied versions 1.2.0 of DBGEN and QGEN were used for this TPC-H benchmark.

3.3 Generating Values for Substitution Parameters

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

The supplied QGEN version 1.2.0 was used to generate the substitution parameters.

3.4 Query Text and Output Data from Qualification Database

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

Appendix C contains the qualification query text and query output.

3.5 Query Substitution Parameters and Seeds Used

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

Appendix 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.

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

3.7 Source Code of Refresh Functions

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

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

4. Clause 3 Database System Properties

4.1 ACID Properties

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

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

4.2 Atomicity

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

4.2.1 Completed Transaction

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

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

4.2.2 Aborted Transaction

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

1. The total price from the ORDERS table and the extended price from the LINEITEM table were retrieved for a randomly selected order key.
2. The ACID Transaction was performed using the order key from step 1. The transaction was stopped prior to the commit.
3. The ACID Transaction was ROLLED BACK.
4. The total price from the ORDERS table and the extended price from the LINEITEM table were retrieved for the same order key. It was verified that the appropriate rows had not been changed.

4.3 Consistency

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

4.3.1 Consistency Test

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

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

4.4 Isolation

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

4.4.1 Read-Write Conflict with Commit

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

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

4.4.2 Read-Write Conflict with Rollback

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

1. An ACID Transaction was started for a randomly selected O_KEY, L_KEY, and DELTA. The ACID Transaction was suspended prior to ROLLBACK.
2. An ACID Query was started for the same O_KEY used in step 1. The ACID Query did not see the uncommitted changes made by the ACID Transaction.
3. The ACID Transaction was ROLLED BACK.
4. The ACID Query completed.

4.4.3 Write-Write Conflict with Commit

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

1. An ACID Transaction, T1, was started for a randomly selected O_KEY, L_KEY, and DELTA. T1 was suspended prior to COMMIT.
2. Another ACID Transaction, T2, was started using the same O_KEY and L_KEY and a randomly selected DELTA.
3. T2 waited.
4. T1 was allowed to COMMIT and T2 completed.

5. It was verified that T2.L_EXTENDEDPRICE = T1.L_EXTENDEDPRICE + (DELTA1*(T1.L_EXTENDEDPRICE/T1.L_QUANTITY))

4.4.4 Write-Write Conflict with Rollback

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

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

4.4.5 Concurrent Progress of Read and Write Transactions

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

1. An ACID Transaction, T1, was started for a randomly selected O_KEY, L_KEY, and DELTA. T1 was suspended prior to ROLLBACK.
2. Another Transaction, T2, was started which did the following:

   For random values of PS_PARTKEY and PS_SUPPKEY, all columns of the PARTSUPP table for which PS_PARTKEY and PS_SUPPKEY are equal, are returned.
3. T2 completed.
4. T1 was allowed to COMMIT.
5. It was verified that appropriate rows in ORDERS, LINEITEM and HISTORY tables were changed.

4.4.6 Read-Only Query Conflict with Update Transaction

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

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

4.5 Durability

The SUT must guarantee durability: the ability to preserve the effects of committed transactions and insure database consistency after recovery from any one of the failures listed in Clause 3.5.3.

4.5.1 Failure of a Durable Medium

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

The disks containing TPC-H 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.

4.5.2 System Crash

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

The system crash and memory failure tests were combined. Power to the server was turned off by flipping breakers at the main electrical panel 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 successfully.

4.5.3 Memory Failure

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

See section 4.5.2.

5. Clause 4 Scaling and Database Population

5.1 Ending Cardinality of Tables

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

Table	Rows
Orders	4,500,000,000
Lineitem	18,000,048,306
Customer	450,000,000
Part	600,000,000
Supplier	30,000,000
Partsupp	2,400,000,000
Nation	25
Region	5

5.2 Distribution of Tables and Logs Across Media

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

- All tables and indexes were mirrored and striped logically across 396 disks in the A5200 arrays.
- The 18 disks of the T3 arrays were used for the redo logs. Each group of 9 disks was configured as RAID0 and mirrored with the remaining 9 disks.
- The temp tablespace was striped across 396 disks in the A5200 arrays but not mirrored.

For more details refer to disk configuration section in Appendix B.

5.3 Database partition/replication mapping

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

The database was not replicated.

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

5.4 RAID Feature

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

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

5.5 Modifications to the DBGEN

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

The supplied DBGEN version 1.2.0 was used to generate the database population for this benchmark.

5.6 Database Load Time

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

The database load time was 8 hours 19 minutes.

5.7 Data Storage Ratio

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

The data storage ratio is computed from the following information:

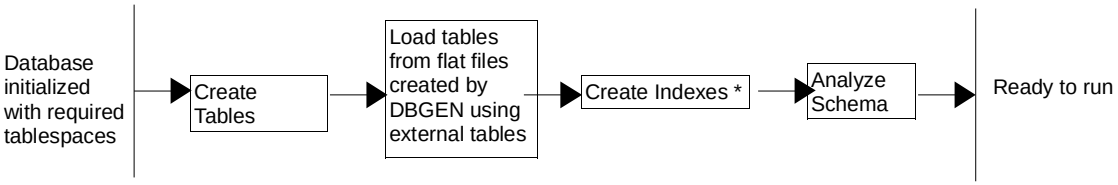
** Disk manufacturer definition of one GB is 10^9 bytes*
***In this calculation one GB is defined as 2^30 bytes*

Disk Type	# Of Disks	Space Per Disk*	Sub-Total Disk Space**
A5200	396	73.4GB	27,070.2 GB
T300	18	36.4GB	610.2 GB
S1	2	36.4GB	67.8 GB
		Total Space	**27,748.2 GB**
		Data Storage Ratio	**9.25**

5.8 Database Load Mechanism Details and Illustration

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

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



* Analyze index performed during index creation

5.9 Qualification Database Configuration

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

The qualification database used identical scripts to create and load the data with adjustments for the size difference.

6. Clause 5 Performance Metrics and Execution Rules

6.1 System Activity Between Load and Performance Tests

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

1. Auditor requested queries were run against the database to verify the correctness of the load

All scripts and queries used are included in Appendix F

6.2 Steps in the Power Test

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

The following steps were used to implement the power test:

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

6.3 Timing Intervals for Each Query and Refresh Functions

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

The power test timing intervals are:

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8a	Q9	Q10	Q11	Q12
Stream 00	828.0	254.3	95.1	65.5	455.2	62.6	423.2	277.1	1126.4	165.5	227.9	344.3

	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	RF1	RF2
Stream 00	725.1	76.2	125.3	273.4	408.0	2090.3	458.5	231.9	1762.2	261.6	402.4	330.8

6.4 Number of Streams for the Throughput Test

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

Eight streams were used for the throughput test.

6.5 Start and End Date/Times for Each Query Stream

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

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

6.6 Total Elapsed Time of the Measurement Interval

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

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

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

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

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

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

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

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

6.9 Performance Metrics

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

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

6.10 The Performance Metric and Numerical Quantities from Both Runs

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

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

Run ID	QppH@3000GB	QthH@3000GB	QphH@3000GB
Run 1	35,779.5	24,255.1	29,440.8
Run 3	34,714.4	24,139.6	28,948.1
Difference	3.07%	0.48%	1.70%

6.11 System Activity Between Performance Tests

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

There was no activity on the SUT between run1 and run2. run2 failed due to an IO error which resulted in the Temporary tablespace becoming unavailable. The IO failure was resolved, the system was rebooted , the database brought up and run3 was started. Run3 ran successfully to completion.

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 and Throughput Test are performed by a shell script called runTPCpt. 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 qexecpl.c 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.

Following the Power Test, QGEN is again called with the subsequent 7 stream ids to generate new QET for each Throughput Test. qexecpl.c is called simultaneously for all 7 streams to execute the queries as above.Then the update_stream.sh script is called to run all 7 update pairs to finish the throughput run.

7.2 Implementation-Specific Layer

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

Query execution text generated by QGEN is picked up by the ISL program which submits the query to the SUT.

The ISL program (qexecpl.c) utilizes the Oracle Call Interface (OCI) to communicate with the Oracle database on the SUT. EQTs directly generated by QGEN are read and submitted to the SUT via the ISL program (qexecpl.c) 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 in Section 5.3.7 of the TPC-H benchmark specification.

The Update Functions use external tables to load data from flat files. Oracle9i's parallel insert and delete functionality was used to perform the Update Functions, selecting data from the temporary tables.

7.3 Profile-Directed Optimization

If profile-directed optimization as described in Clause 5.2.9 is used, such use must be disclosed.

Profile-directed optimization was not used.

8. Clause 7 Pricing

8.1 Hardware and Software Used

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

Refer to the Executive Summary.

8.2 Total Three Year Price

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

The total 5-year price of the configuration is $5,335,742. For details of pricing, see the second page of the Executive Summary.

The following generally available discounts to any buyer with like conditions were applied to the priced configuration:

- a 8% Sun support volume discount
- a 3% Sun support yearly pre-payment discount
- a 40% discount from list for Sun supplied system components

8.3 Availability Date

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

All Hardware and Software components, except Solaris 9 Update 3, are available immediately. Solaris 9 Update 3 will be available April 30, 2003.

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 9 and Oracle9i Parameters

This Appendix contains Solaris kernel parameters and environment variables and Oracle initialization parameters.

```
================================================
```

Oracle 9i R2 Parameters

(altered from default)

```
================================================
```

init_audit.ora

```
sort_area_size                    =16384
statistics_level                  = basic
audit_trail                       = FALSE
compatible                        = 9.2.0
control_files                     =
/dev/vx/rdsk/photon0/cntrl_1
db_block_checksum                 = false
db_block_size                     = 16384
db_cache_size                     = 18g
db_file_multiblock_read_count     = 128
db_files                          = 1023
db_name                           = inst1
db_writer_processes               = 20
dml_locks                         = 80000
enqueue_resources                 = 50000
global_names                      = FALSE
java_pool_size                    = 0
large_pool_size                   = 8g
log_buffer                        = 67108864
log_parallelism                   = 4
log_checkpoints_to_alert          = TRUE
nls_date_format                   = YYYY-MM-DD
open_cursors                      = 1024
optimizer_features_enable         = 9.2.0.1
optimizer_mode                    = CHOOSE
parallel_adaptive_multi_user      = TRUE
parallel_automatic_tuning         = TRUE
parallel_execution_message_size   = 16384
parallel_max_servers              = 1000
parallel_min_servers              = 1000
parallel_threads_per_cpu          = 3
pga_aggregate_target              = 70g
processes                         = 1024
query_rewrite_enabled             = TRUE
replication_dependency_tracking   = FALSE
sessions                          = 1024
shared_pool_size                  = 200M
transaction_auditing              = FALSE
undo_management                   = AUTO
cpu_count                         = 72
```

```
================================================
```

Oracle Environment Variables

```
================================================
export KIT_DIR=/export/btmp/home/oracle/kit
export SCHEMA_DIR=$KIT_DIR/schema
export PERL=/opt/PERL/bin/perl
export BUMPX_DIR=$KIT_DIR/bumpx
export BUMPX_OUT=$KIT_DIR/bumpx
export UTILS=$KIT_DIR/utils
export TEST_DB=$KIT_DIR/acid
export QUAL_DB=$TEST_DB
export DBGEN=$KIT_DIR/dbgen
export ACID_DIR=$KIT_DIR/acid
export QEXEC=$KIT_DIR/utils
export QUERIES=$KIT_DIR/queries
export ANSWERS=$KIT_DIR/answers
export
ANS2VAL=/export/btmp/home/oracle/kit/acid/answers
export ACID_OUT=$QUAL_DB/acid_out
export DSS_CONFIG=$DBGEN
export DSS_QUERY=$KIT_DIR/queries
export DSS_PATH=$ADE_VIEW_ROOT
export MAINT=$KIT_DIR/maintenance
export CC=cc
export FRAME=$KIT_DIR/frame
export REGR_TEST=$KIT_DIR/internal/regression_test
export SCALE_FACTOR=3000
export UPDATE_DOP_INS=32
export UPDATE_DOP_DEL=144
================================================
```

Solaris Parameters

(altered from default)

```
================================================
```

/etc/system

```
================================================
set shmsys:shminfo_shmmax=0xffffffffffffffff
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=1024
set shmsys:shminfo_shmseg=500

set semsys:seminfo_semmap=8388608
set semsys:seminfo_semmni=4096
set semsys:seminfo_semmns=8388608
set semsys:seminfo_semmnu=4096
set semsys:seminfo_semmsl=2048
set semsys:seminfo_semume=2048
set semsys:seminfo_semopm=100
set semsys:seminfo_semvmx=32767

set msgsys:msginfo_msgmap=2048
set msgsys:msginfo_msgmax=8192
set msgsys:msginfo_msgmnb=16384
set msgsys:msginfo_msgssz=32
set msgsys:msginfo_msgtql=2048
set msgsys:msginfo_msgseg=32767

set maxpgio=131072
set maxphys=4194304
set autoup=1800
set tune_t_fsflushr=10
set bufhwm=8000
set segspt_minfree=16000
set seg_pwindow=28311552
set p_hashsize=131072
set seg_pmaxqlen=128
set segmap_percent=2

set ge:ge_tx_bcopy_max = 60
set ge:ge_tx_fastdvma_min = 59
set ge:ge_dmaburst_mode = 0x1

set lgrp_mem_default_policy = 0x1
set kernel_cage_enable = 0x0
set kcpc_counts_include_idle=0
set ce:ce_taskq_disable=1
set ssd:ssd_max_throttle = 96
```

Appendix B. Programs and Scripts

==========

bumpx.pl

==========

```
#!/bin/perl

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

sub readfile {
    $matchon = 0;
    $contin = 0;
    $pkey = "";
    $pval = "";
    open (INFILE, "$infile");
  WLOOP:
    while ($line = <INFILE>)
    {
        $line = $line."\n" if $line !~ /\n/;
        study $line;
        if ($line =~ /^\*matchon/)
        {
            $matchon = 1;
            next WLOOP;
        }
        if ($line =~ /^\*matchoff/)
        {
            $matchon = 0;
            next WLOOP;
        }
        if ($matchon == 1)
        {
            &dump0($line) if $docreate;
            next WLOOP;
        }
        next WLOOP if $line =~ /^\s*\#/;
        next WLOOP if $line =~ /^\s*\n/;
        if ($contin)
        {
            if ($line =~ /(.*)\&\s*\n/) # still
continuing (changed \ to &)
            {
                $pval = $pval . $1;
                next WLOOP;
            }
            $line =~ /(.*)\s*\n/; # reached the end
            $pval = $pval . $1;
            $pval =~ s/\?/$ENV{'ORACLE_HOME'}/g;
            &key_exists($pkey);
            if ($result!=1)
            {
                print "Parameter $pkey does not
exist.\nBailing out!\n";
                exit (0);
            }
            $params{$pkey} = $pval;
            $contin = 0;
            $pkey = "";
            $pval = "";
            next WLOOP;
        }
        else
        {
            if ($line =~ /\s*(\S+)\s*=\s*(.*)\&\s*\n/)
            {
                $pkey = $1;
                $pval = $2;
                $contin = 1;
                next WLOOP;
            }
            if ($line =~ /\s*(\S+)\s*=\s*\n/)
            {
                undef($params{$1});
                next WLOOP;
            }
            if ($line !~
/\s*(\S+)\s*=\s*((\S+)|(\S+.*\S+))\s*\n/)
            {
                print "Bad line: $line";
                next WLOOP;
            }
            $tkey = $1;
            $tval = $2;
            $tval =~ s/\?/$ENV{'ORACLE_HOME'}/g;
           if ($tval =~ /\$/) {  # a $ sign at the
beginning of the contents
                                  # of the parameter
value results in a environment
                                   # variable lookup
and substitution
                $tenv = $tval;
                $tval =~ s/\$//g;
                if ($unix) {
```

```
                    $tenv =~
s/(.*)\/\$(.*)\/(.*)/\$$2/g;
                }
                $tenvval = `echo $tenv`;
                $tenv =~ s/\$//g;
                chop($tenvval);
                if ($tenvval=~/^\n/) {
                    print "bumpx error: Environment
variable $tval not defined!\nbailing out...\n";
                    exit(1);
                }
                $tval =~ s/$tenv/$tenvval/g;
            }

            &key_exists($tkey);
            if ($result!=1)
            {
                print "Parameter $tkey does not
exists.\nBailing out!\n";
                exit (0);
            }
             if ($tval =~ /^\[.*\]$/) { # found a
program to execute
                $tval =~ s/\[//; $tval =~ s/\]//;
                print "$tkey:$tval is a program\n";
                open (POUT,">/tmp/bumpx_prg.pl");
                print POUT "#!/usr/local/bin/perl\n";
                print POUT $tval;
                 close POUT;
                 `chmod ogu+x /tmp/bumpx_prg.pl`;
                $result=`/tmp/bumpx_prg.pl`;
                 print $result;
                 $tval=$result;
            }
            $params{$tkey} = $tval;
        }
    }
    close (INFILE);
}

sub doexecute { # First, do preprocessing stuff
    print "Execution pass begun." if $verbose;
    open (INFILE, $outfile);
  WLOOP1:
    while ($line = <INFILE>)
    {
        study $line;
        next WLOOP1 if $line =~ /^\s*\#/;
        next WLOOP1 if $line =~ /^\s*\n/;
        if ($line =~ /^%b-preproc/)
        {
            $insection = 1;
            next WLOOP1;
        }
        next WLOOP1 if ($insection != 1);
        if ($line =~ /^%e-preproc/)
        {
            $insection = 0;
            $commands{$shortcmd} = $longcmd if defined
$shortcmd;
            last WLOOP1;
        }
        if ($line =~ /^\*/)
        {
            $commands{$shortcmd} = $longcmd if defined
$shortcmd;
            $line =~ /^(\*.*\S+)\s*\n$/;
            $shortcmd = $1;
            $longcmd = "";
            next WLOOP1;
        }
        if ($line =~ /^\\/)
        {
            $line =~ /\\(.*\n)/;
            $longcmd = $longcmd . $1;
            next WLOOP1;
        }
        print "Illegal entry in preproc stage:\n
$line";
    }
    close (INFILE);

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

sub execute
{
    $cmd = shift(@_);
    if ($cmd)
```

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

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

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

```
                $cmd =~
s/phase\#.lst/$listdir$phase\_$phase_cmd_num.lst/;
                ++$phase_cmd_num;
                print TMPFILE $cmd;
                close(TMPFILE);
                system("cat $tmpfile >>
$listdir$phase.lst");
                system ("sh $tmpfile");
                if ($?) {
                    print "system\'d command
failed:\nsh $tmpfile\nreason: $? ($!)\n";
                    print "Please check the contents in
the shell script.\n";
                    exit(-1);
                }
                open(TMPFILE, ">$tmpfile");
             }
        }  ## NT support End
    }
}

sub usage
{
    print "Usage:\n";
    print "This is a lite version of bumpx.pl.  It can
only be used to execute a .dat file\n";
    print " $0 [-o outfile] [-p phaselist] [-t
type]\n";
    print "   -o : intermediary file to be created
and/or used\n";
    print "        defaults to bumpx.dat in
\$BUMPX_DIR or \$CWD\n";
    print "   -p : list of phases to
create/execute\n";
    print "        phaselist is a comma separated list
of phases in order\n";
    print "        possible phases are:\n";
    print "          sdgen = seed file generation\n";
    print "          dbgen = data flat file
generation\n";
    print "          plcre = NT raw partition and
links creation\n";
    print "          dbcre = database creation\n";
    print "          shutd = shutdown database (on all
instances)\n";
    print "          start = startup database (on all
instances)\n";
    print "          sccre = schema creation\n";
    print "          sctso = schema creation
(tablespaces only)\n";
    print "          scuto = schema creation (user and
tables only)\n";
    print "          scuvo = schema creation (views
only)\n";
    print "          dapop = data population\n";
    print "          ixcre = index creation (including
constraints)\n";
    print "          anlyz = analyze objects\n";
    print "          chob  = change parameters of
objects\n";
    print "          expln = create explain plans\n";
    print "          query = run and time queries\n";
    print "        defaults to $defphases\n";
    print "   -t : type of benchmark\n";
    print "        enables benchmark-specific
defaults\n";
    print "        current possibilties are:
$allbmtypes\n";
    print "        defaults to tpcd\n";
    print "   -s : run silent (no parameter checking
is done)\n";
    print "\n";
    print "Examples:\n";
    print " $0 -p dapop\n";
    print "   Executes data population phase of
intermediary file bumpx.dat.\n";
    print "\n";
    print "$error\n";
    exit(-1);
}
```

==========

3tera.66way.dat

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

%b-preproc

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

*load
\sqlldr {}

*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-dbcre
*bgon=1
############################################################
######################
# Database Creation Phase
*sql
{
shutdown abort;
}
*wait
# creating database and initial rollback segment
*sql
{
startup pfile= ?/dbs/init_inst1.ora nomount;
create database
  controlfile reuse
  logfile
'/export/home/oracle/oracle9i/dbs/datafiles/t3log_1'
size 31000m reuse,
          '/export/home/oracle/oracle9i/dbs/datafiles/
t3log_2' size 31000m reuse
  datafile
'/export/home/oracle/oracle9i/dbs/datafiles/sys_1'
size 1000m reuse
  maxdatafiles 10000
  maxinstances 2
  undo tablespace ts_undo
       datafile
'/export/home/oracle/oracle9i/dbs/datafiles/undo_1'
size 19997m reuse
;
}
*wait
# building data dictionary
*sql
{
set termout off
set echo off
spool /tmp/cat
@?/rdbms/admin/catalog.sql;
@?/rdbms/admin/catparr.sql;
@?/rdbms/admin/catproc.sql;
@?/rdbms/admin/utlxplan.sql;
connect system/manager
@/ora9i/sqlplus/admin/pupbld.sql;
}
*wait
*bgoff
%e-dbcre
%b-sctso
*bgon=144
############################################################
######################
# Schema Creation Phase - datafiles only (no tables or
```

```
users)
# creating data tablespaces, datafiles
# creating tpcd's ts_default tablespace
*sql
{
drop tablespace ts_default including contents;
create tablespace ts_default
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/default_1'
size 9900m reuse
extent management local autoallocate ;
}
# creating tpcd's ts_s tablespace
*sql
{
drop tablespace ts_s including contents;
create tablespace ts_s
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/s_1' size
852m reuse
extent management local autoallocate nologging ;
}
# creating tpcd's ts_c tablespace
*sql
{
drop tablespace ts_c including contents;
create tablespace ts_c
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/c_1' size
14303m reuse
extent management local autoallocate nologging ;
}
# creating tpcd's ts_ps tablespace
*sql
{
drop tablespace ts_ps including contents;
create tablespace ts_ps
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_1' size
19931m reuse
extent management local autoallocate nologging ;
}
# creating tpcd's ts_p tablespace
*sql
{
drop tablespace ts_p including contents;
create tablespace ts_p
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/p_1' size
13511m reuse
extent management local autoallocate nologging ;
}
# creating tpcd's ts_o1 tablespace
*sql
{
drop tablespace ts_o1 including contents;
create tablespace ts_o1
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_1' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o2 tablespace
*sql
{
drop tablespace ts_o2 including contents;
create tablespace ts_o2
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_2' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o3 tablespace
*sql
{
drop tablespace ts_o3 including contents;
create tablespace ts_o3
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_3' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o4 tablespace
*sql
{
drop tablespace ts_o4 including contents;
create tablespace ts_o4
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_4' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o5 tablespace
*sql
{
drop tablespace ts_o5 including contents;
create tablespace ts_o5
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_5' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o6 tablespace
*sql
{
drop tablespace ts_o6 including contents;
create tablespace ts_o6
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_6' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o7 tablespace
*sql
{
drop tablespace ts_o7 including contents;
create tablespace ts_o7
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_7' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o8 tablespace
*sql
{
drop tablespace ts_o8 including contents;
create tablespace ts_o8
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_8' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o9 tablespace
*sql
{
drop tablespace ts_o9 including contents;
create tablespace ts_o9
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_9' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o10 tablespace
*sql
{
drop tablespace ts_o10 including contents;
create tablespace ts_o10
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_10' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o11 tablespace
*sql
{
drop tablespace ts_o11 including contents;
create tablespace ts_o11
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_11' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o12 tablespace
*sql
{
drop tablespace ts_o12 including contents;
create tablespace ts_o12
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_12' size
8368m reuse
extent management local autoallocate nologging  ;
}
```

```
# creating tpcd's ts_o13 tablespace
*sql
{
drop tablespace ts_o13 including contents;
create tablespace ts_o13
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_13' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o14 tablespace
*sql
{
drop tablespace ts_o14 including contents;
create tablespace ts_o14
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_14' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o15 tablespace
*sql
{
drop tablespace ts_o15 including contents;
create tablespace ts_o15
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_15' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o16 tablespace
*sql
{
drop tablespace ts_o16 including contents;
create tablespace ts_o16
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_16' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o17 tablespace
*sql
{
drop tablespace ts_o17 including contents;
create tablespace ts_o17
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_17' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o18 tablespace
*sql
{
drop tablespace ts_o18 including contents;
create tablespace ts_o18
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_18' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o19 tablespace
*sql
{
drop tablespace ts_o19 including contents;
create tablespace ts_o19
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_19' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o20 tablespace
*sql
{
drop tablespace ts_o20 including contents;
create tablespace ts_o20
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_20' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o21 tablespace
*sql
{
drop tablespace ts_o21 including contents;
create tablespace ts_o21
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_21' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o22 tablespace
*sql
{
drop tablespace ts_o22 including contents;
create tablespace ts_o22
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_22' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o23 tablespace
*sql
{
drop tablespace ts_o23 including contents;
create tablespace ts_o23
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_23' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o24 tablespace
*sql
{
drop tablespace ts_o24 including contents;
create tablespace ts_o24
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_24' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o25 tablespace
*sql
{
drop tablespace ts_o25 including contents;
create tablespace ts_o25
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_25' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o26 tablespace
*sql
{
drop tablespace ts_o26 including contents;
create tablespace ts_o26
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_26' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o27 tablespace
*sql
{
drop tablespace ts_o27 including contents;
create tablespace ts_o27
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_27' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o28 tablespace
*sql
{
drop tablespace ts_o28 including contents;
create tablespace ts_o28
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_28' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o29 tablespace
*sql
{
drop tablespace ts_o29 including contents;
create tablespace ts_o29
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_29' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o30 tablespace
*sql
```

```
{
drop tablespace ts_o30 including contents;
create tablespace ts_o30
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_30' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o31 tablespace
*sql
{
drop tablespace ts_o31 including contents;
create tablespace ts_o31
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_31' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o32 tablespace
*sql
{
drop tablespace ts_o32 including contents;
create tablespace ts_o32
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_32' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o33 tablespace
*sql
{
drop tablespace ts_o33 including contents;
create tablespace ts_o33
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_33' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o34 tablespace
*sql
{
drop tablespace ts_o34 including contents;
create tablespace ts_o34
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_34' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o35 tablespace
*sql
{
drop tablespace ts_o35 including contents;
create tablespace ts_o35
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_35' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o36 tablespace
*sql
{
drop tablespace ts_o36 including contents;
create tablespace ts_o36
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_36' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o37 tablespace
*sql
{
drop tablespace ts_o37 including contents;
create tablespace ts_o37
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_37' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o38 tablespace
*sql
{
drop tablespace ts_o38 including contents;
create tablespace ts_o38
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_38' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o39 tablespace
*sql
{
drop tablespace ts_o39 including contents;
create tablespace ts_o39
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_39' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o40 tablespace
*sql
{
drop tablespace ts_o40 including contents;
create tablespace ts_o40
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_40' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o41 tablespace
*sql
{
drop tablespace ts_o41 including contents;
create tablespace ts_o41
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_41' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o42 tablespace
*sql
{
drop tablespace ts_o42 including contents;
create tablespace ts_o42
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_42' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o43 tablespace
*sql
{
drop tablespace ts_o43 including contents;
create tablespace ts_o43
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_43' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o44 tablespace
*sql
{
drop tablespace ts_o44 including contents;
create tablespace ts_o44
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_44' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o45 tablespace
*sql
{
drop tablespace ts_o45 including contents;
create tablespace ts_o45
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_45' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o46 tablespace
*sql
{
drop tablespace ts_o46 including contents;
create tablespace ts_o46
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_46' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o47 tablespace
*sql
{
drop tablespace ts_o47 including contents;
```

```
create tablespace ts_o47
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_47' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o48 tablespace
*sql
{
drop tablespace ts_o48 including contents;
create tablespace ts_o48
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_48' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o49 tablespace
*sql
{
drop tablespace ts_o49 including contents;
create tablespace ts_o49
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_49' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o50 tablespace
*sql
{
drop tablespace ts_o50 including contents;
create tablespace ts_o50
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_50' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o51 tablespace
*sql
{
drop tablespace ts_o51 including contents;
create tablespace ts_o51
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_51' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o52 tablespace
*sql
{
drop tablespace ts_o52 including contents;
create tablespace ts_o52
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_52' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o53 tablespace
*sql
{
drop tablespace ts_o53 including contents;
create tablespace ts_o53
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_53' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o54 tablespace
*sql
{
drop tablespace ts_o54 including contents;
create tablespace ts_o54
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_54' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o55 tablespace
*sql
{
drop tablespace ts_o55 including contents;
create tablespace ts_o55
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_55' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o56 tablespace
*sql
{
drop tablespace ts_o56 including contents;
create tablespace ts_o56
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_56' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o57 tablespace
*sql
{
drop tablespace ts_o57 including contents;
create tablespace ts_o57
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_57' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o58 tablespace
*sql
{
drop tablespace ts_o58 including contents;
create tablespace ts_o58
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_58' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o59 tablespace
*sql
{
drop tablespace ts_o59 including contents;
create tablespace ts_o59
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_59' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o60 tablespace
*sql
{
drop tablespace ts_o60 including contents;
create tablespace ts_o60
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_60' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o61 tablespace
*sql
{
drop tablespace ts_o61 including contents;
create tablespace ts_o61
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_61' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o62 tablespace
*sql
{
drop tablespace ts_o62 including contents;
create tablespace ts_o62
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_62' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o63 tablespace
*sql
{
drop tablespace ts_o63 including contents;
create tablespace ts_o63
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_63' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o64 tablespace
*sql
{
drop tablespace ts_o64 including contents;
create tablespace ts_o64
datafile
```

```
'/export/home/oracle/oracle9i/dbs/datafiles/o_64' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o65 tablespace
*sql
{
drop tablespace ts_o65 including contents;
create tablespace ts_o65
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_65' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o66 tablespace
*sql
{
drop tablespace ts_o66 including contents;
create tablespace ts_o66
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_66' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o67 tablespace
*sql
{
drop tablespace ts_o67 including contents;
create tablespace ts_o67
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_67' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o68 tablespace
*sql
{
drop tablespace ts_o68 including contents;
create tablespace ts_o68
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_68' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o69 tablespace
*sql
{
drop tablespace ts_o69 including contents;
create tablespace ts_o69
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_69' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o70 tablespace
*sql
{
drop tablespace ts_o70 including contents;
create tablespace ts_o70
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_70' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o71 tablespace
*sql
{
drop tablespace ts_o71 including contents;
create tablespace ts_o71
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_71' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o72 tablespace
*sql
{
drop tablespace ts_o72 including contents;
create tablespace ts_o72
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_72' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o73 tablespace
*sql
{
drop tablespace ts_o73 including contents;
create tablespace ts_o73
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_73' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o74 tablespace
*sql
{
drop tablespace ts_o74 including contents;
create tablespace ts_o74
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_74' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o75 tablespace
*sql
{
drop tablespace ts_o75 including contents;
create tablespace ts_o75
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_75' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o76 tablespace
*sql
{
drop tablespace ts_o76 including contents;
create tablespace ts_o76
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_76' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o77 tablespace
*sql
{
drop tablespace ts_o77 including contents;
create tablespace ts_o77
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_77' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o78 tablespace
*sql
{
drop tablespace ts_o78 including contents;
create tablespace ts_o78
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_78' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o79 tablespace
*sql
{
drop tablespace ts_o79 including contents;
create tablespace ts_o79
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_79' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o80 tablespace
*sql
{
drop tablespace ts_o80 including contents;
create tablespace ts_o80
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_80' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o81 tablespace
*sql
{
drop tablespace ts_o81 including contents;
create tablespace ts_o81
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_81' size
8368m reuse
```

```
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o82 tablespace
*sql
{
drop tablespace ts_o82 including contents;
create tablespace ts_o82
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_82' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o83 tablespace
*sql
{
drop tablespace ts_o83 including contents;
create tablespace ts_o83
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_83' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_o84 tablespace
*sql
{
drop tablespace ts_o84 including contents;
create tablespace ts_o84
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/o_84' size
8368m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l1 tablespace
*sql
{
drop tablespace ts_l1 including contents;
create tablespace ts_l1
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_1' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l2 tablespace
*sql
{
drop tablespace ts_l2 including contents;
create tablespace ts_l2
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_2' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l3 tablespace
*sql
{
drop tablespace ts_l3 including contents;
create tablespace ts_l3
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_3' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l4 tablespace
*sql
{
drop tablespace ts_l4 including contents;
create tablespace ts_l4
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_4' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l5 tablespace
*sql
{
drop tablespace ts_l5 including contents;
create tablespace ts_l5
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_5' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l6 tablespace
*sql
{
drop tablespace ts_l6 including contents;
create tablespace ts_l6
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_6' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l7 tablespace
*sql
{
drop tablespace ts_l7 including contents;
create tablespace ts_l7
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_7' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l8 tablespace
*sql
{
drop tablespace ts_l8 including contents;
create tablespace ts_l8
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_8' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l9 tablespace
*sql
{
drop tablespace ts_l9 including contents;
create tablespace ts_l9
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_9' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l10 tablespace
*sql
{
drop tablespace ts_l10 including contents;
create tablespace ts_l10
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_10' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l11 tablespace
*sql
{
drop tablespace ts_l11 including contents;
create tablespace ts_l11
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_11' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l12 tablespace
*sql
{
drop tablespace ts_l12 including contents;
create tablespace ts_l12
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_12' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l13 tablespace
*sql
{
drop tablespace ts_l13 including contents;
create tablespace ts_l13
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_13' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l14 tablespace
*sql
{
drop tablespace ts_l14 including contents;
create tablespace ts_l14
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_14' size
40000m reuse
extent management local autoallocate nologging  ;
}
```

```
# creating tpcd's ts_l15 tablespace
*sql
{
drop tablespace ts_l15 including contents;
create tablespace ts_l15
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_15' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l16 tablespace
*sql
{
drop tablespace ts_l16 including contents;
create tablespace ts_l16
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_16' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l17 tablespace
*sql
{
drop tablespace ts_l17 including contents;
create tablespace ts_l17
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_17' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l18 tablespace
*sql
{
drop tablespace ts_l18 including contents;
create tablespace ts_l18
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_18' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l19 tablespace
*sql
{
drop tablespace ts_l19 including contents;
create tablespace ts_l19
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_19' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l20 tablespace
*sql
{
drop tablespace ts_l20 including contents;
create tablespace ts_l20
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_20' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l21 tablespace
*sql
{
drop tablespace ts_l21 including contents;
create tablespace ts_l21
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_21' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l22 tablespace
*sql
{
drop tablespace ts_l22 including contents;
create tablespace ts_l22
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_22' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l23 tablespace
*sql
{
drop tablespace ts_l23 including contents;
create tablespace ts_l23
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_23' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l24 tablespace
*sql
{
drop tablespace ts_l24 including contents;
create tablespace ts_l24
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_24' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l25 tablespace
*sql
{
drop tablespace ts_l25 including contents;
create tablespace ts_l25
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_25' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l26 tablespace
*sql
{
drop tablespace ts_l26 including contents;
create tablespace ts_l26
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_26' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l27 tablespace
*sql
{
drop tablespace ts_l27 including contents;
create tablespace ts_l27
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_27' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l28 tablespace
*sql
{
drop tablespace ts_l28 including contents;
create tablespace ts_l28
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_28' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l29 tablespace
*sql
{
drop tablespace ts_l29 including contents;
create tablespace ts_l29
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_29' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l30 tablespace
*sql
{
drop tablespace ts_l30 including contents;
create tablespace ts_l30
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_30' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l31 tablespace
*sql
{
drop tablespace ts_l31 including contents;
create tablespace ts_l31
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_31' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l32 tablespace
*sql
```

```
{
drop tablespace ts_l32 including contents;
create tablespace ts_l32
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_32' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l33 tablespace
*sql
{
drop tablespace ts_l33 including contents;
create tablespace ts_l33
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_33' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l34 tablespace
*sql
{
drop tablespace ts_l34 including contents;
create tablespace ts_l34
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_34' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l35 tablespace
*sql
{
drop tablespace ts_l35 including contents;
create tablespace ts_l35
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_35' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l36 tablespace
*sql
{
drop tablespace ts_l36 including contents;
create tablespace ts_l36
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_36' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l37 tablespace
*sql
{
drop tablespace ts_l37 including contents;
create tablespace ts_l37
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_37' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l38 tablespace
*sql
{
drop tablespace ts_l38 including contents;
create tablespace ts_l38
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_38' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l39 tablespace
*sql
{
drop tablespace ts_l39 including contents;
create tablespace ts_l39
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_39' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l40 tablespace
*sql
{
drop tablespace ts_l40 including contents;
create tablespace ts_l40
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_40' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l41 tablespace
*sql
{
drop tablespace ts_l41 including contents;
create tablespace ts_l41
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_41' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l42 tablespace
*sql
{
drop tablespace ts_l42 including contents;
create tablespace ts_l42
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_42' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l43 tablespace
*sql
{
drop tablespace ts_l43 including contents;
create tablespace ts_l43
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_43' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l44 tablespace
*sql
{
drop tablespace ts_l44 including contents;
create tablespace ts_l44
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_44' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l45 tablespace
*sql
{
drop tablespace ts_l45 including contents;
create tablespace ts_l45
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_45' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l46 tablespace
*sql
{
drop tablespace ts_l46 including contents;
create tablespace ts_l46
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_46' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l47 tablespace
*sql
{
drop tablespace ts_l47 including contents;
create tablespace ts_l47
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_47' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l48 tablespace
*sql
{
drop tablespace ts_l48 including contents;
create tablespace ts_l48
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_48' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l49 tablespace
*sql
{
drop tablespace ts_l49 including contents;
```

```
create tablespace ts_l49
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_49' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l50 tablespace
*sql
{
drop tablespace ts_l50 including contents;
create tablespace ts_l50
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_50' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l51 tablespace
*sql
{
drop tablespace ts_l51 including contents;
create tablespace ts_l51
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_51' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l52 tablespace
*sql
{
drop tablespace ts_l52 including contents;
create tablespace ts_l52
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_52' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l53 tablespace
*sql
{
drop tablespace ts_l53 including contents;
create tablespace ts_l53
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_53' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l54 tablespace
*sql
{
drop tablespace ts_l54 including contents;
create tablespace ts_l54
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_54' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l55 tablespace
*sql
{
drop tablespace ts_l55 including contents;
create tablespace ts_l55
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_55' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l56 tablespace
*sql
{
drop tablespace ts_l56 including contents;
create tablespace ts_l56
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_56' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l57 tablespace
*sql
{
drop tablespace ts_l57 including contents;
create tablespace ts_l57
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_57' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l58 tablespace
*sql
{
drop tablespace ts_l58 including contents;
create tablespace ts_l58
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_58' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l59 tablespace
*sql
{
drop tablespace ts_l59 including contents;
create tablespace ts_l59
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_59' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l60 tablespace
*sql
{
drop tablespace ts_l60 including contents;
create tablespace ts_l60
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_60' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l61 tablespace
*sql
{
drop tablespace ts_l61 including contents;
create tablespace ts_l61
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_61' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l62 tablespace
*sql
{
drop tablespace ts_l62 including contents;
create tablespace ts_l62
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_62' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l63 tablespace
*sql
{
drop tablespace ts_l63 including contents;
create tablespace ts_l63
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_63' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l64 tablespace
*sql
{
drop tablespace ts_l64 including contents;
create tablespace ts_l64
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_64' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l65 tablespace
*sql
{
drop tablespace ts_l65 including contents;
create tablespace ts_l65
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_65' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l66 tablespace
*sql
{
drop tablespace ts_l66 including contents;
create tablespace ts_l66
datafile
```

```
'/export/home/oracle/oracle9i/dbs/datafiles/l_66' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l67 tablespace
*sql
{
drop tablespace ts_l67 including contents;
create tablespace ts_l67
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_67' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l68 tablespace
*sql
{
drop tablespace ts_l68 including contents;
create tablespace ts_l68
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_68' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l69 tablespace
*sql
{
drop tablespace ts_l69 including contents;
create tablespace ts_l69
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_69' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l70 tablespace
*sql
{
drop tablespace ts_l70 including contents;
create tablespace ts_l70
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_70' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l71 tablespace
*sql
{
drop tablespace ts_l71 including contents;
create tablespace ts_l71
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_71' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l72 tablespace
*sql
{
drop tablespace ts_l72 including contents;
create tablespace ts_l72
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_72' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l73 tablespace
*sql
{
drop tablespace ts_l73 including contents;
create tablespace ts_l73
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_73' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l74 tablespace
*sql
{
drop tablespace ts_l74 including contents;
create tablespace ts_l74
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_74' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l75 tablespace
*sql
{
drop tablespace ts_l75 including contents;
create tablespace ts_l75
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_75' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l76 tablespace
*sql
{
drop tablespace ts_l76 including contents;
create tablespace ts_l76
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_76' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l77 tablespace
*sql
{
drop tablespace ts_l77 including contents;
create tablespace ts_l77
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_77' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l78 tablespace
*sql
{
drop tablespace ts_l78 including contents;
create tablespace ts_l78
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_78' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l79 tablespace
*sql
{
drop tablespace ts_l79 including contents;
create tablespace ts_l79
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_79' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l80 tablespace
*sql
{
drop tablespace ts_l80 including contents;
create tablespace ts_l80
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_80' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l81 tablespace
*sql
{
drop tablespace ts_l81 including contents;
create tablespace ts_l81
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_81' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l82 tablespace
*sql
{
drop tablespace ts_l82 including contents;
create tablespace ts_l82
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_82' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l83 tablespace
*sql
{
drop tablespace ts_l83 including contents;
create tablespace ts_l83
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_83' size
40000m reuse
```

```
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_l84 tablespace
*sql
{
drop tablespace ts_l84 including contents;
create tablespace ts_l84
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/l_84' size
40000m reuse
extent management local autoallocate nologging  ;
}
# creating tpcd's ts_i_lorderkey tablespace
*sql
{
drop tablespace ts_i_lorderkey including contents;
create tablespace ts_i_lorderkey
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_l_orderk
ey_1' size 42671m reuse
extent management local autoallocate nologging ;
}
# creating tpcd's ts_i_oorderkey tablespace
*sql
{
drop tablespace ts_i_oorderkey including contents;
create tablespace ts_i_oorderkey
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_o_orderk
ey_1' size 25399m reuse
extent management local autoallocate nologging ;
}
# creating tpcd's ts_i_ccustkey tablespace
*sql
{
drop tablespace ts_i_ccustkey including contents;
create tablespace ts_i_ccustkey
datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_c_custke
y_1' size 10360m reuse
extent management local autoallocate nologging ;
}
# creating tpcd's ts_temp tablespace
*sql
{
drop tablespace ts_temp including contents;
create temporary tablespace ts_temp
tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_1'
size 28445m reuse
extent management local uniform size 100M ;
}
*wait
*wait
# adding tpcd's ts_undo datafiles
*sql
{
alter tablespace ts_undo
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/undo_2'
size 19997m reuse;
}
*sql
{
alter tablespace ts_undo
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/undo_3'
size 19997m reuse;
}
*sql
{
alter tablespace ts_undo
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/undo_4'
size 19997m reuse;
}
*sql
{
alter tablespace ts_undo
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/undo_5'
size 19997m reuse;
}
*sql
{
alter tablespace ts_undo
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/undo_6'
size 19997m reuse;
}
*sql
{
alter tablespace ts_undo
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/undo_7'
size 19997m reuse;
}
*sql
{
alter tablespace ts_undo
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/undo_8'
size 19997m reuse;
}
*sql
{
alter tablespace ts_undo
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/undo_9'
size 19997m reuse;
}
*sql
{
alter tablespace ts_undo
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/undo_10'
size 19997m reuse;
}
*sql
{
alter tablespace ts_undo
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/undo_11'
size 19997m reuse;
}
*sql
{
alter tablespace ts_undo
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/undo_12'
size 19997m reuse;
}
# adding tpcd's ts_s datafiles
*sql
{
alter tablespace ts_s
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/s_2' size
852m reuse;
}
*sql
{
alter tablespace ts_s
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/s_3' size
852m reuse;
}
*sql
{
alter tablespace ts_s
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/s_4' size
852m reuse;
}
*sql
{
alter tablespace ts_s
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/s_5' size
852m reuse;
}
*sql
{
alter tablespace ts_s
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/s_6' size
852m reuse;
}
# adding tpcd's ts_c datafiles
*sql
{
alter tablespace ts_c
```

```
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/c_2' size
14303m reuse;
}
*sql
{
alter tablespace ts_c
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/c_3' size
14303m reuse;
}
*sql
{
alter tablespace ts_c
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/c_4' size
14303m reuse;
}
*sql
{
alter tablespace ts_c
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/c_5' size
14303m reuse;
}
*sql
{
alter tablespace ts_c
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/c_6' size
14303m reuse;
}
# adding tpcd's ts_ps datafiles
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_2' size
19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_3' size
19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_4' size
19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_5' size
19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_6' size
19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_7' size
19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_8' size
19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_9' size
19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_10'
size 19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_11'
size 19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_12'
size 19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_13'
size 19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_14'
size 19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_15'
size 19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_16'
size 19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_17'
size 19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_18'
size 19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_19'
size 19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_20'
size 19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_21'
size 19931m reuse;
```

```
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_22'
size 19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_23'
size 19931m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/ps_24'
size 19931m reuse;
}
# adding tpcd's ts_p datafiles
*sql
{
alter tablespace ts_p
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/p_2' size
13511m reuse;
}
*sql
{
alter tablespace ts_p
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/p_3' size
13511m reuse;
}
*sql
{
alter tablespace ts_p
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/p_4' size
13511m reuse;
}
*sql
{
alter tablespace ts_p
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/p_5' size
13511m reuse;
}
*sql
{
alter tablespace ts_p
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/p_6' size
13511m reuse;
}
# adding tpcd's ts_i_lorderkey datafiles
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_l_orderk
ey_2' size 42671m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_l_orderk
ey_3' size 42671m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_l_orderk
ey_4' size 42671m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_l_orderk
ey_5' size 42671m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_l_orderk
ey_6' size 42671m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_l_orderk
ey_7' size 42671m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_l_orderk
ey_8' size 42671m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_l_orderk
ey_9' size 42671m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_l_orderk
ey_10' size 42671m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_l_orderk
ey_11' size 42671m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_l_orderk
ey_12' size 42671m reuse;
}
# adding tpcd's ts_i_oorderkey datafiles
*sql
{
alter tablespace ts_i_oorderkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_o_orderk
ey_2' size 25399m reuse;
}
*sql
{
alter tablespace ts_i_oorderkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_o_orderk
ey_3' size 25399m reuse;
}
*sql
{
alter tablespace ts_i_oorderkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_o_orderk
ey_4' size 25399m reuse;
}
*sql
{
alter tablespace ts_i_oorderkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_o_orderk
ey_5' size 25399m reuse;
}
*sql
{
alter tablespace ts_i_oorderkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_o_orderk
ey_6' size 25399m reuse;
}
```

```
# adding tpcd's ts_i_ccustkey datafiles
*sql
{
alter tablespace ts_i_ccustkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_c_custke
y_2' size 10360m reuse;
}
*sql
{
alter tablespace ts_i_ccustkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_c_custke
y_3' size 10360m reuse;
}
*sql
{
alter tablespace ts_i_ccustkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_c_custke
y_4' size 10360m reuse;
}
*sql
{
alter tablespace ts_i_ccustkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_c_custke
y_5' size 10360m reuse;
}
*sql
{
alter tablespace ts_i_ccustkey
  add datafile
'/export/home/oracle/oracle9i/dbs/datafiles/i_c_custke
y_6' size 10360m reuse;
}
# adding tpcd's ts_temp datafiles
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_2'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_3'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_4'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_5'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_6'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_7'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_8'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_9'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_10'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_11'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_12'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_13'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_14'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_15'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_16'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_17'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_18'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_19'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_20'
size 28445m reuse;
}
*sql
{
```

```
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_21'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_22'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_23'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_24'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_25'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_26'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_27'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_28'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_29'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_30'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_31'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_32'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_33'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_34'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_35'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_36'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_37'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_38'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_39'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_40'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_41'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_42'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_43'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_44'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_45'
size 28445m reuse;
```

```
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_46'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_47'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_48'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_49'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_50'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_51'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_52'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_53'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_54'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_55'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_56'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_57'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_58'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_59'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_60'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_61'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_62'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_63'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_64'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_65'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_66'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_67'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_68'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_69'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
```

```
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_70'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_71'
size 28445m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle9i/dbs/datafiles/temp_72'
size 28445m reuse;
}
*wait
*wait
*bgoff
%e-sctso
%b-dapop
*bgon=1
#########################################################
######################
# Schema Creation Phase - User and Tables
#  AND  Database Population Phase
#
# creating tpcd user
*sql
{
drop user tpcd cascade;
grant DBA
to tpcd identified by tpcd;
}
*wait
*sql
{
connect tpcd/tpcd;
drop directory data_dir;
create directory data_dir as '/flat1';
}
*sql
{
connect tpcd/tpcd;
drop table l_et;
create table l_et(
    l_shipdate          date ,
    l_orderkey          number ,
    l_discount          number ,
    l_extendedprice     number ,
    l_suppkey           number ,
    l_quantity          number ,
    l_returnflag        char(1) ,
    l_partkey           number ,
    l_linestatus        char(1) ,
    l_tax               number ,
    l_commitdate        date ,
    l_receiptdate       date ,
    l_shipmode          char(10) ,
    l_linenumber        number ,
    l_shipinstruct      char(25) ,
    l_comment           varchar(44)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                records delimited by newline
                badfile 'l_et.bad'
                logfile 'l_et.log'
                fields terminated by '|'
                missing field values are null
        )
        location (
        'lineitem.tbl.1',
        'lineitem.tbl.2',
        'lineitem.tbl.3',
        'lineitem.tbl.4',
        'lineitem.tbl.5',
        'lineitem.tbl.6',
        'lineitem.tbl.7',
        'lineitem.tbl.8',
        'lineitem.tbl.9',
        'lineitem.tbl.10',
        'lineitem.tbl.11',
        'lineitem.tbl.12',
        'lineitem.tbl.13',
        'lineitem.tbl.14',
        'lineitem.tbl.15',
        'lineitem.tbl.16',
        'lineitem.tbl.17',
        'lineitem.tbl.18',
        'lineitem.tbl.19',
        'lineitem.tbl.20',
        'lineitem.tbl.21',
        'lineitem.tbl.22',
        'lineitem.tbl.23',
        'lineitem.tbl.24',
        'lineitem.tbl.25',
        'lineitem.tbl.26',
        'lineitem.tbl.27',
        'lineitem.tbl.28',
        'lineitem.tbl.29',
        'lineitem.tbl.30',
        'lineitem.tbl.31',
        'lineitem.tbl.32',
        'lineitem.tbl.33',
        'lineitem.tbl.34',
        'lineitem.tbl.35',
        'lineitem.tbl.36',
        'lineitem.tbl.37',
        'lineitem.tbl.38',
        'lineitem.tbl.39',
        'lineitem.tbl.40',
        'lineitem.tbl.41',
        'lineitem.tbl.42',
        'lineitem.tbl.43',
        'lineitem.tbl.44',
        'lineitem.tbl.45',
        'lineitem.tbl.46',
        'lineitem.tbl.47',
        'lineitem.tbl.48',
        'lineitem.tbl.49',
        'lineitem.tbl.50',
        'lineitem.tbl.51',
        'lineitem.tbl.52',
        'lineitem.tbl.53',
        'lineitem.tbl.54',
        'lineitem.tbl.55',
        'lineitem.tbl.56',
        'lineitem.tbl.57',
        'lineitem.tbl.58',
        'lineitem.tbl.59',
        'lineitem.tbl.60',
        'lineitem.tbl.61',
        'lineitem.tbl.62',
        'lineitem.tbl.63',
        'lineitem.tbl.64',
        'lineitem.tbl.65',
        'lineitem.tbl.66',
        'lineitem.tbl.67',
        'lineitem.tbl.68',
        'lineitem.tbl.69',
        'lineitem.tbl.70',
        'lineitem.tbl.71',
        'lineitem.tbl.72',
        'lineitem.tbl.73',
        'lineitem.tbl.74',
        'lineitem.tbl.75',
        'lineitem.tbl.76',
        'lineitem.tbl.77',
        'lineitem.tbl.78',
        'lineitem.tbl.79',
        'lineitem.tbl.80',
        'lineitem.tbl.81',
        'lineitem.tbl.82',
        'lineitem.tbl.83',
        'lineitem.tbl.84'
        ))
reject limit unlimited;
}
*sql
{
connect tpcd/tpcd;
drop table o_et;
create table o_et(
    o_orderdate         date ,
    o_orderkey          number ,
    o_custkey           number ,
```

```
    o_orderpriority      char(15) ,
    o_shippriority       number ,
    o_clerk              char(15) ,
    o_orderstatus        char(1) ,
    o_totalprice         number ,
    o_comment            varchar(79)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                records delimited by newline
                badfile 'o_et.bad'
                logfile 'o_et.log'
                fields terminated by '|'
                missing field values are null
        )
        location (
        'orders.tbl.1',
        'orders.tbl.2',
        'orders.tbl.3',
        'orders.tbl.4',
        'orders.tbl.5',
        'orders.tbl.6',
        'orders.tbl.7',
        'orders.tbl.8',
        'orders.tbl.9',
        'orders.tbl.10',
        'orders.tbl.11',
        'orders.tbl.12',
        'orders.tbl.13',
        'orders.tbl.14',
        'orders.tbl.15',
        'orders.tbl.16',
        'orders.tbl.17',
        'orders.tbl.18',
        'orders.tbl.19',
        'orders.tbl.20',
        'orders.tbl.21',
        'orders.tbl.22',
        'orders.tbl.23',
        'orders.tbl.24',
        'orders.tbl.25',
        'orders.tbl.26',
        'orders.tbl.27',
        'orders.tbl.28',
        'orders.tbl.29',
        'orders.tbl.30',
        'orders.tbl.31',
        'orders.tbl.32',
        'orders.tbl.33',
        'orders.tbl.34',
        'orders.tbl.35',
        'orders.tbl.36',
        'orders.tbl.37',
        'orders.tbl.38',
        'orders.tbl.39',
        'orders.tbl.40',
        'orders.tbl.41',
        'orders.tbl.42',
        'orders.tbl.43',
        'orders.tbl.44',
        'orders.tbl.45',
        'orders.tbl.46',
        'orders.tbl.47',
        'orders.tbl.48',
        'orders.tbl.49',
        'orders.tbl.50',
        'orders.tbl.51',
        'orders.tbl.52',
        'orders.tbl.53',
        'orders.tbl.54',
        'orders.tbl.55',
        'orders.tbl.56',
        'orders.tbl.57',
        'orders.tbl.58',
        'orders.tbl.59',
        'orders.tbl.60',
        'orders.tbl.61',
        'orders.tbl.62',
        'orders.tbl.63',
        'orders.tbl.64',
        'orders.tbl.65',
        'orders.tbl.66',
        'orders.tbl.67',
        'orders.tbl.68',
        'orders.tbl.69',
        'orders.tbl.70',
        'orders.tbl.71',
        'orders.tbl.72',
        'orders.tbl.73',
        'orders.tbl.74',
        'orders.tbl.75',
        'orders.tbl.76',
        'orders.tbl.77',
        'orders.tbl.78',
        'orders.tbl.79',
        'orders.tbl.80',
        'orders.tbl.81',
        'orders.tbl.82',
        'orders.tbl.83',
        'orders.tbl.84'
        ))
reject limit unlimited;
}
*sql
{
connect tpcd/tpcd;
drop table ps_et;
create table ps_et(
    ps_partkey           number ,
    ps_suppkey           number ,
    ps_supplycost        number ,
    ps_availqty          number ,
    ps_comment           varchar(199)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                records delimited by newline
                badfile 'ps_et.bad'
                logfile 'ps_et.log'
                fields terminated by '|'
                missing field values are null
        )
        location (
        'partsupp.tbl.1',
        'partsupp.tbl.2',
        'partsupp.tbl.3',
        'partsupp.tbl.4',
        'partsupp.tbl.5',
        'partsupp.tbl.6',
        'partsupp.tbl.7',
        'partsupp.tbl.8',
        'partsupp.tbl.9',
        'partsupp.tbl.10',
        'partsupp.tbl.11',
        'partsupp.tbl.12',
        'partsupp.tbl.13',
        'partsupp.tbl.14',
        'partsupp.tbl.15',
        'partsupp.tbl.16',
        'partsupp.tbl.17',
        'partsupp.tbl.18',
        'partsupp.tbl.19',
        'partsupp.tbl.20',
        'partsupp.tbl.21',
        'partsupp.tbl.22',
        'partsupp.tbl.23',
        'partsupp.tbl.24',
        'partsupp.tbl.25',
        'partsupp.tbl.26',
        'partsupp.tbl.27',
        'partsupp.tbl.28',
        'partsupp.tbl.29',
        'partsupp.tbl.30',
        'partsupp.tbl.31',
        'partsupp.tbl.32',
        'partsupp.tbl.33',
        'partsupp.tbl.34',
        'partsupp.tbl.35',
        'partsupp.tbl.36'
        ))
reject limit unlimited;
}
*sql
{
connect tpcd/tpcd;
drop table p_et;
```

```
create table p_et(
    p_partkey           number ,
    p_type              varchar(25) ,
    p_size              number ,
    p_brand             char(10) ,
    p_name              varchar(55) ,
    p_container         char(10) ,
    p_mfgr              char(25) ,
    p_retailprice       number ,
    p_comment           varchar(23)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                records delimited by newline
                badfile 'p_et.bad'
                logfile 'p_et.log'
                fields terminated by '|'
                missing field values are null
        )
        location (
        'part.tbl.1',
        'part.tbl.2',
        'part.tbl.3',
        'part.tbl.4',
        'part.tbl.5',
        'part.tbl.6',
        'part.tbl.7',
        'part.tbl.8'
))
reject limit unlimited;
}
*sql
{
connect tpcd/tpcd;
drop table c_et;
create table c_et(
    c_custkey           number ,
    c_mktsegment        char(10) ,
    c_nationkey         number ,
    c_name              varchar(25) ,
    c_address           varchar(40) ,
    c_phone             char(15) ,
    c_acctbal           number ,
    c_comment           varchar(117)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                records delimited by newline
                badfile 'c_et.bad'
                logfile 'c_et.log'
                fields terminated by '|'
                missing field values are null
        )
        location (
        'customer.tbl.1',
        'customer.tbl.2',
        'customer.tbl.3',
        'customer.tbl.4',
        'customer.tbl.5',
        'customer.tbl.6',
        'customer.tbl.7',
        'customer.tbl.8'
))
reject limit unlimited;
}
*sql
{
connect tpcd/tpcd;
drop table s_et;
create table s_et(
    s_suppkey           number ,
    s_nationkey         number ,
    s_comment           varchar(101) ,
    s_name              char(25) ,
    s_address           varchar(40) ,
    s_phone             char(15) ,
    s_acctbal           number
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                records delimited by newline
                badfile 's_et.bad'
                logfile 's_et.log'
                fields terminated by '|'
                missing field values are null
        )
        location (
        'supplier.tbl'
        ))
reject limit unlimited;
}
*sql
{
connect tpcd/tpcd;
drop table n_et;
create table n_et(
    n_nationkey         number ,
    n_name              char(25) ,
    n_regionkey         number ,
    n_comment           varchar(152)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                records delimited by newline
                badfile 'n_et.bad'
                logfile 'n_et.log'
                fields terminated by '|'
                missing field values are null
        )
        location (
        'nation.tbl'))
reject limit unlimited;
}
*sql
{
connect tpcd/tpcd;
drop table r_et;
create table r_et(
    r_regionkey         number ,
    r_name              char(25) ,
    r_comment           varchar(152)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                records delimited by newline
                badfile 'r_et.bad'
                logfile 'r_et.log'
                fields terminated by '|'
                missing field values are null
        )
        location (
        'region.tbl'))
reject limit unlimited;
}
*sql
{
connect tpcd/tpcd;
alter table l_et parallel 144;
alter table o_et parallel 144;
alter table ps_et parallel 144;
alter table p_et parallel 144;
alter table c_et parallel 144;
alter table s_et parallel 144;
}
# altering tpcd's default and temporary tablespace
*sql
{
alter user tpcd default tablespace ts_default;
alter user tpcd temporary tablespace ts_temp;
}
*sql
{
rem connect tpcd/tpcd
rem @?/rdbms/admin/utlxplan.sql;
}
*wait
*sql
```

```
{
!date
connect tpcd/tpcd;
set timing on
set echo on

rem drop table lineitem;
create table lineitem(
    l_shipdate          ,
    l_orderkey          NOT NULL,
    l_discount          NOT NULL,
    l_extendedprice     NOT NULL,
    l_suppkey           NOT NULL,
    l_quantity          NOT NULL,
    l_returnflag        ,
    l_partkey           NOT NULL,
    l_linestatus        ,
    l_tax               NOT NULL,
    l_commitdate        ,
    l_receiptdate       ,
    l_shipmode          ,
    l_linenumber        NOT NULL,
    l_shipinstruct      ,
    l_comment
)
pctfree 1
pctused 99
initrans 10
storage (initial 1500m freelist groups 4 freelists 84)
parallel 144
nologging
partition by range (l_shipdate)
subpartition by hash(l_partkey)
subpartitions 16
(
partition item1 values less than
(to_date('1992-01-01','YYYY-MM-DD'))
tablespace ts_l1
,
partition item2 values less than
(to_date('1992-02-01','YYYY-MM-DD'))
tablespace ts_l2
,
partition item3 values less than
(to_date('1992-03-01','YYYY-MM-DD'))
tablespace ts_l3
,
partition item4 values less than
(to_date('1992-04-01','YYYY-MM-DD'))
tablespace ts_l4
,
partition item5 values less than
(to_date('1992-05-01','YYYY-MM-DD'))
tablespace ts_l5
,
partition item6 values less than
(to_date('1992-06-01','YYYY-MM-DD'))
tablespace ts_l6
,
partition item7 values less than
(to_date('1992-07-01','YYYY-MM-DD'))
tablespace ts_l7
,
partition item8 values less than
(to_date('1992-08-01','YYYY-MM-DD'))
tablespace ts_l8
,
partition item9 values less than
(to_date('1992-09-01','YYYY-MM-DD'))
tablespace ts_l9
,
partition item10 values less than
(to_date('1992-10-01','YYYY-MM-DD'))
tablespace ts_l10
,
partition item11 values less than
(to_date('1992-11-01','YYYY-MM-DD'))
tablespace ts_l11
,
partition item12 values less than
(to_date('1992-12-01','YYYY-MM-DD'))
tablespace ts_l12
,
partition item13 values less than
(to_date('1993-01-01','YYYY-MM-DD'))
tablespace ts_l13
,
partition item14 values less than
(to_date('1993-02-01','YYYY-MM-DD'))
tablespace ts_l14
,
partition item15 values less than
(to_date('1993-03-01','YYYY-MM-DD'))
tablespace ts_l15
,
partition item16 values less than
(to_date('1993-04-01','YYYY-MM-DD'))
tablespace ts_l16
,
partition item17 values less than
(to_date('1993-05-01','YYYY-MM-DD'))
tablespace ts_l17
,
partition item18 values less than
(to_date('1993-06-01','YYYY-MM-DD'))
tablespace ts_l18
,
partition item19 values less than
(to_date('1993-07-01','YYYY-MM-DD'))
tablespace ts_l19
,
partition item20 values less than
(to_date('1993-08-01','YYYY-MM-DD'))
tablespace ts_l20
,
partition item21 values less than
(to_date('1993-09-01','YYYY-MM-DD'))
tablespace ts_l21
,
partition item22 values less than
(to_date('1993-10-01','YYYY-MM-DD'))
tablespace ts_l22
,
partition item23 values less than
(to_date('1993-11-01','YYYY-MM-DD'))
tablespace ts_l23
,
partition item24 values less than
(to_date('1993-12-01','YYYY-MM-DD'))
tablespace ts_l24
,
partition item25 values less than
(to_date('1994-01-01','YYYY-MM-DD'))
tablespace ts_l25
,
partition item26 values less than
(to_date('1994-02-01','YYYY-MM-DD'))
tablespace ts_l26
,
partition item27 values less than
(to_date('1994-03-01','YYYY-MM-DD'))
tablespace ts_l27
,
partition item28 values less than
(to_date('1994-04-01','YYYY-MM-DD'))
tablespace ts_l28
,
partition item29 values less than
(to_date('1994-05-01','YYYY-MM-DD'))
tablespace ts_l29
,
partition item30 values less than
(to_date('1994-06-01','YYYY-MM-DD'))
tablespace ts_l30
,
partition item31 values less than
(to_date('1994-07-01','YYYY-MM-DD'))
tablespace ts_l31
,
partition item32 values less than
(to_date('1994-08-01','YYYY-MM-DD'))
tablespace ts_l32
,
partition item33 values less than
(to_date('1994-09-01','YYYY-MM-DD'))
tablespace ts_l33
,
partition item34 values less than
(to_date('1994-10-01','YYYY-MM-DD'))
tablespace ts_l34
,
partition item35 values less than
```

```
(to_date('1994-11-01','YYYY-MM-DD'))
tablespace ts_l35
,
partition item36 values less than
(to_date('1994-12-01','YYYY-MM-DD'))
tablespace ts_l36
,
partition item37 values less than
(to_date('1995-01-01','YYYY-MM-DD'))
tablespace ts_l37
,
partition item38 values less than
(to_date('1995-02-01','YYYY-MM-DD'))
tablespace ts_l38
,
partition item39 values less than
(to_date('1995-03-01','YYYY-MM-DD'))
tablespace ts_l39
,
partition item40 values less than
(to_date('1995-04-01','YYYY-MM-DD'))
tablespace ts_l40
,
partition item41 values less than
(to_date('1995-05-01','YYYY-MM-DD'))
tablespace ts_l41
,
partition item42 values less than
(to_date('1995-06-01','YYYY-MM-DD'))
tablespace ts_l42
,
partition item43 values less than
(to_date('1995-07-01','YYYY-MM-DD'))
tablespace ts_l43
,
partition item44 values less than
(to_date('1995-08-01','YYYY-MM-DD'))
tablespace ts_l44
,
partition item45 values less than
(to_date('1995-09-01','YYYY-MM-DD'))
tablespace ts_l45
,
partition item46 values less than
(to_date('1995-10-01','YYYY-MM-DD'))
tablespace ts_l46
,
partition item47 values less than
(to_date('1995-11-01','YYYY-MM-DD'))
tablespace ts_l47
,
partition item48 values less than
(to_date('1995-12-01','YYYY-MM-DD'))
tablespace ts_l48
,
partition item49 values less than
(to_date('1996-01-01','YYYY-MM-DD'))
tablespace ts_l49
,
partition item50 values less than
(to_date('1996-02-01','YYYY-MM-DD'))
tablespace ts_l50
,
partition item51 values less than
(to_date('1996-03-01','YYYY-MM-DD'))
tablespace ts_l51
,
partition item52 values less than
(to_date('1996-04-01','YYYY-MM-DD'))
tablespace ts_l52
,
partition item53 values less than
(to_date('1996-05-01','YYYY-MM-DD'))
tablespace ts_l53
,
partition item54 values less than
(to_date('1996-06-01','YYYY-MM-DD'))
tablespace ts_l54
,
partition item55 values less than
(to_date('1996-07-01','YYYY-MM-DD'))
tablespace ts_l55
,
partition item56 values less than
(to_date('1996-08-01','YYYY-MM-DD'))
tablespace ts_l56
,
partition item57 values less than
(to_date('1996-09-01','YYYY-MM-DD'))
tablespace ts_l57
,
partition item58 values less than
(to_date('1996-10-01','YYYY-MM-DD'))
tablespace ts_l58
,
partition item59 values less than
(to_date('1996-11-01','YYYY-MM-DD'))
tablespace ts_l59
,
partition item60 values less than
(to_date('1996-12-01','YYYY-MM-DD'))
tablespace ts_l60
,
partition item61 values less than
(to_date('1997-01-01','YYYY-MM-DD'))
tablespace ts_l61
,
partition item62 values less than
(to_date('1997-02-01','YYYY-MM-DD'))
tablespace ts_l62
,
partition item63 values less than
(to_date('1997-03-01','YYYY-MM-DD'))
tablespace ts_l63
,
partition item64 values less than
(to_date('1997-04-01','YYYY-MM-DD'))
tablespace ts_l64
,
partition item65 values less than
(to_date('1997-05-01','YYYY-MM-DD'))
tablespace ts_l65
,
partition item66 values less than
(to_date('1997-06-01','YYYY-MM-DD'))
tablespace ts_l66
,
partition item67 values less than
(to_date('1997-07-01','YYYY-MM-DD'))
tablespace ts_l67
,
partition item68 values less than
(to_date('1997-08-01','YYYY-MM-DD'))
tablespace ts_l68
,
partition item69 values less than
(to_date('1997-09-01','YYYY-MM-DD'))
tablespace ts_l69
,
partition item70 values less than
(to_date('1997-10-01','YYYY-MM-DD'))
tablespace ts_l70
,
partition item71 values less than
(to_date('1997-11-01','YYYY-MM-DD'))
tablespace ts_l71
,
partition item72 values less than
(to_date('1997-12-01','YYYY-MM-DD'))
tablespace ts_l72
,
partition item73 values less than
(to_date('1998-01-01','YYYY-MM-DD'))
tablespace ts_l73
,
partition item74 values less than
(to_date('1998-02-01','YYYY-MM-DD'))
tablespace ts_l74
,
partition item75 values less than
(to_date('1998-03-01','YYYY-MM-DD'))
tablespace ts_l75
,
partition item76 values less than
(to_date('1998-04-01','YYYY-MM-DD'))
tablespace ts_l76
,
partition item77 values less than
(to_date('1998-05-01','YYYY-MM-DD'))
tablespace ts_l77
,
partition item78 values less than
```

```
(to_date('1998-06-01','YYYY-MM-DD'))
tablespace ts_l78
,
partition item79 values less than
(to_date('1998-07-01','YYYY-MM-DD'))
tablespace ts_l79
,
partition item80 values less than
(to_date('1998-08-01','YYYY-MM-DD'))
tablespace ts_l80
,
partition item81 values less than
(to_date('1998-09-01','YYYY-MM-DD'))
tablespace ts_l81
,
partition item82 values less than
(to_date('1998-10-01','YYYY-MM-DD'))
tablespace ts_l82
,
partition item83 values less than
(to_date('1998-11-01','YYYY-MM-DD'))
tablespace ts_l83
,
partition item84 values less than (MAXVALUE)
tablespace ts_l84
)
as select * from l_et;
}
*wait
*sql
{
connect tpcd/tpcd;
set timing on
set echo on

!date
rem drop table orders;
create table orders(
    o_orderdate          ,
    o_orderkey           NOT NULL,
    o_custkey            NOT NULL,
    o_orderpriority      ,
    o_shippriority       ,
    o_clerk              ,
    o_orderstatus        ,
    o_totalprice         ,
    o_comment            )
pctfree 1
pctused 99
initrans 10
storage (initial 400m freelist groups 4 freelists 84)
parallel 144
nologging
partition by range (o_orderdate)
subpartition by hash(o_custkey)
subpartitions 16
(
partition ord1 values less than (to_date('1992-01-01',
'YYYY-MM-DD'))
tablespace ts_o1
,
partition ord2 values less than (to_date('1992-02-01',
'YYYY-MM-DD'))
tablespace ts_o2
,
partition ord3 values less than (to_date('1992-03-01',
'YYYY-MM-DD'))
tablespace ts_o3
,
partition ord4 values less than (to_date('1992-04-01',
'YYYY-MM-DD'))
tablespace ts_o4
,
partition ord5 values less than (to_date('1992-05-01',
'YYYY-MM-DD'))
tablespace ts_o5
,
partition ord6 values less than (to_date('1992-06-01',
'YYYY-MM-DD'))
tablespace ts_o6
,
partition ord7 values less than (to_date('1992-07-01',
'YYYY-MM-DD'))
tablespace ts_o7
,
partition ord8 values less than (to_date('1992-08-01',
'YYYY-MM-DD'))
tablespace ts_o8
,
partition ord9 values less than (to_date('1992-09-01',
'YYYY-MM-DD'))
tablespace ts_o9
,
partition ord10 values less than
(to_date('1992-10-01','YYYY-MM-DD'))
tablespace ts_o10
,
partition ord11 values less than
(to_date('1992-11-01','YYYY-MM-DD'))
tablespace ts_o11
,
partition ord12 values less than
(to_date('1992-12-01','YYYY-MM-DD'))
tablespace ts_o12
,
partition ord13 values less than
(to_date('1993-01-01','YYYY-MM-DD'))
tablespace ts_o13
,
partition ord14 values less than
(to_date('1993-02-01','YYYY-MM-DD'))
tablespace ts_o14
,
partition ord15 values less than
(to_date('1993-03-01','YYYY-MM-DD'))
tablespace ts_o15
,
partition ord16 values less than
(to_date('1993-04-01','YYYY-MM-DD'))
tablespace ts_o16
,
partition ord17 values less than
(to_date('1993-05-01','YYYY-MM-DD'))
tablespace ts_o17
,
partition ord18 values less than
(to_date('1993-06-01','YYYY-MM-DD'))
tablespace ts_o18
,
partition ord19 values less than
(to_date('1993-07-01','YYYY-MM-DD'))
tablespace ts_o19
,
partition ord20 values less than
(to_date('1993-08-01','YYYY-MM-DD'))
tablespace ts_o20
,
partition ord21 values less than
(to_date('1993-09-01','YYYY-MM-DD'))
tablespace ts_o21
,
partition ord22 values less than
(to_date('1993-10-01','YYYY-MM-DD'))
tablespace ts_o22
,
partition ord23 values less than
(to_date('1993-11-01','YYYY-MM-DD'))
tablespace ts_o23
,
partition ord24 values less than
(to_date('1993-12-01','YYYY-MM-DD'))
tablespace ts_o24
,
partition ord25 values less than
(to_date('1994-01-01','YYYY-MM-DD'))
tablespace ts_o25
,
partition ord26 values less than
(to_date('1994-02-01','YYYY-MM-DD'))
tablespace ts_o26
,
partition ord27 values less than
(to_date('1994-03-01','YYYY-MM-DD'))
tablespace ts_o27
,
partition ord28 values less than
(to_date('1994-04-01','YYYY-MM-DD'))
tablespace ts_o28
,
partition ord29 values less than
(to_date('1994-05-01','YYYY-MM-DD'))
tablespace ts_o29
```

```
,
partition ord30 values less than
(to_date('1994-06-01','YYYY-MM-DD'))
tablespace ts_o30
,
partition ord31 values less than
(to_date('1994-07-01','YYYY-MM-DD'))
tablespace ts_o31
,
partition ord32 values less than
(to_date('1994-08-01','YYYY-MM-DD'))
tablespace ts_o32
,
partition ord33 values less than
(to_date('1994-09-01','YYYY-MM-DD'))
tablespace ts_o33
,
partition ord34 values less than
(to_date('1994-10-01','YYYY-MM-DD'))
tablespace ts_o34
,
partition ord35 values less than
(to_date('1994-11-01','YYYY-MM-DD'))
tablespace ts_o35
,
partition ord36 values less than
(to_date('1994-12-01','YYYY-MM-DD'))
tablespace ts_o36
,
partition ord37 values less than
(to_date('1995-01-01','YYYY-MM-DD'))
tablespace ts_o37
,
partition ord38 values less than
(to_date('1995-02-01','YYYY-MM-DD'))
tablespace ts_o38
,
partition ord39 values less than
(to_date('1995-03-01','YYYY-MM-DD'))
tablespace ts_o39
,
partition ord40 values less than
(to_date('1995-04-01','YYYY-MM-DD'))
tablespace ts_o40
,
partition ord41 values less than
(to_date('1995-05-01','YYYY-MM-DD'))
tablespace ts_o41
,
partition ord42 values less than
(to_date('1995-06-01','YYYY-MM-DD'))
tablespace ts_o42
,
partition ord43 values less than
(to_date('1995-07-01','YYYY-MM-DD'))
tablespace ts_o43
,
partition ord44 values less than
(to_date('1995-08-01','YYYY-MM-DD'))
tablespace ts_o44
,
partition ord45 values less than
(to_date('1995-09-01','YYYY-MM-DD'))
tablespace ts_o45
,
partition ord46 values less than
(to_date('1995-10-01','YYYY-MM-DD'))
tablespace ts_o46
,
partition ord47 values less than
(to_date('1995-11-01','YYYY-MM-DD'))
tablespace ts_o47
,
partition ord48 values less than
(to_date('1995-12-01','YYYY-MM-DD'))
tablespace ts_o48
,
partition ord49 values less than
(to_date('1996-01-01','YYYY-MM-DD'))
tablespace ts_o49
,
partition ord50 values less than
(to_date('1996-02-01','YYYY-MM-DD'))
tablespace ts_o50
,
partition ord51 values less than
(to_date('1996-03-01','YYYY-MM-DD'))
tablespace ts_o51
,
partition ord52 values less than
(to_date('1996-04-01','YYYY-MM-DD'))
tablespace ts_o52
,
partition ord53 values less than
(to_date('1996-05-01','YYYY-MM-DD'))
tablespace ts_o53
,
partition ord54 values less than
(to_date('1996-06-01','YYYY-MM-DD'))
tablespace ts_o54
,
partition ord55 values less than
(to_date('1996-07-01','YYYY-MM-DD'))
tablespace ts_o55
,
partition ord56 values less than
(to_date('1996-08-01','YYYY-MM-DD'))
tablespace ts_o56
,
partition ord57 values less than
(to_date('1996-09-01','YYYY-MM-DD'))
tablespace ts_o57
,
partition ord58 values less than
(to_date('1996-10-01','YYYY-MM-DD'))
tablespace ts_o58
,
partition ord59 values less than
(to_date('1996-11-01','YYYY-MM-DD'))
tablespace ts_o59
,
partition ord60 values less than
(to_date('1996-12-01','YYYY-MM-DD'))
tablespace ts_o60
,
partition ord61 values less than
(to_date('1997-01-01','YYYY-MM-DD'))
tablespace ts_o61
,
partition ord62 values less than
(to_date('1997-02-01','YYYY-MM-DD'))
tablespace ts_o62
,
partition ord63 values less than
(to_date('1997-03-01','YYYY-MM-DD'))
tablespace ts_o63
,
partition ord64 values less than
(to_date('1997-04-01','YYYY-MM-DD'))
tablespace ts_o64
,
partition ord65 values less than
(to_date('1997-05-01','YYYY-MM-DD'))
tablespace ts_o65
,
partition ord66 values less than
(to_date('1997-06-01','YYYY-MM-DD'))
tablespace ts_o66
,
partition ord67 values less than
(to_date('1997-07-01','YYYY-MM-DD'))
tablespace ts_o67
,
partition ord68 values less than
(to_date('1997-08-01','YYYY-MM-DD'))
tablespace ts_o68
,
partition ord69 values less than
(to_date('1997-09-01','YYYY-MM-DD'))
tablespace ts_o69
,
partition ord70 values less than
(to_date('1997-10-01','YYYY-MM-DD'))
tablespace ts_o70
,
partition ord71 values less than
(to_date('1997-11-01','YYYY-MM-DD'))
tablespace ts_o71
,
partition ord72 values less than
(to_date('1997-12-01','YYYY-MM-DD'))
tablespace ts_o72
```

```
,
partition ord73 values less than
(to_date('1998-01-01','YYYY-MM-DD'))
tablespace ts_o73
,
partition ord74 values less than
(to_date('1998-02-01','YYYY-MM-DD'))
tablespace ts_o74
,
partition ord75 values less than
(to_date('1998-03-01','YYYY-MM-DD'))
tablespace ts_o75
,
partition ord76 values less than
(to_date('1998-04-01','YYYY-MM-DD'))
tablespace ts_o76
,
partition ord77 values less than
(to_date('1998-05-01','YYYY-MM-DD'))
tablespace ts_o77
,
partition ord78 values less than
(to_date('1998-06-01','YYYY-MM-DD'))
tablespace ts_o78
,
partition ord79 values less than
(to_date('1998-07-01','YYYY-MM-DD'))
tablespace ts_o79
,
partition ord80 values less than
(to_date('1998-08-01','YYYY-MM-DD'))
tablespace ts_o80
,
partition ord81 values less than
(to_date('1998-09-01','YYYY-MM-DD'))
tablespace ts_o81
,
partition ord82 values less than
(to_date('1998-10-01','YYYY-MM-DD'))
tablespace ts_o82
,
partition ord83 values less than
(to_date('1998-11-01','YYYY-MM-DD'))
tablespace ts_o83
,
partition ord84 values less than (MAXVALUE)
tablespace ts_o84
)
as select * from o_et;
}
*wait
*sql
{
connect tpcd/tpcd
set timing on
set echo on

!date
rem drop table partsupp;
create table partsupp(
    ps_partkey          NOT NULL,
    ps_suppkey          NOT NULL,
    ps_supplycost       ,
    ps_availqty         ,
    ps_comment          ,
constraint pk_partkey_suppkey_1 primary
key(ps_partkey, ps_suppkey)
)
organization index tablespace ts_ps
partition by hash (ps_partkey)
partitions 144
compress
pctthreshold 50
storage (initial 1500m)
parallel 144
nologging
as select * from ps_et;
!date
}
*wait
*sql
{
connect tpcd/tpcd
set timing on
set echo on

!date
rem drop table part;
create table part(
    p_partkey           NOT NULL,
    p_type              ,
    p_size              ,
    p_brand             ,
    p_name              ,
    p_container         ,
    p_mfgr              ,
    p_retailprice       ,
    p_comment           )
pctfree 0
pctused 99
tablespace ts_p
storage (freelists 99)
parallel 144
nologging
partition by hash (p_partkey)
partitions 144
as select * from p_et;
}
*wait
*sql
{
connect tpcd/tpcd;
set timing on
set echo on

!date
rem drop table customer;
create table customer(
    c_custkey           NOT NULL,
    c_mktsegment        ,
    c_nationkey         ,
    c_name              ,
    c_address           ,
    c_phone             ,
    c_acctbal           ,
    c_comment           )
pctfree 0
pctused 99
tablespace ts_c
storage (freelists 99)
parallel 144
nologging
partition by hash (c_custkey)
partitions 144
as select * from c_et;
}
*wait
*sql
{
connect tpcd/tpcd;
set timing on
set echo on
rem drop table supplier;
create table supplier(
    s_suppkey           NOT NULL,
    s_nationkey         ,
    s_comment           ,
    s_name              ,
    s_address           ,
    s_phone             ,
    s_acctbal           )
pctfree 0
pctused 99
tablespace ts_s
storage (freelists 99)
parallel 144
nologging
partition by hash (s_suppkey)
partitions 144
as select * from s_et;
}
*wait
*sql
{
connect tpcd/tpcd;
set echo on
set timing on

rem drop table nation;
create table nation(
    n_nationkey         NOT NULL,
    n_name              ,
```

```
    n_regionkey          ,
    n_comment            )
tablespace ts_default
as select * from n_et;

rem drop table region;
create table region(
    r_regionkey          ,
    r_name               ,
    r_comment            )
tablespace ts_default
as select * from r_et;
}
*wait
*bgoff
%e-scuto

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

!date
drop table l_et;
drop table o_et;
drop table ps_et;
drop table p_et;
drop table c_et;
drop table s_et;
drop table n_et;
drop table r_et;
}
*bgoff
%e-dapop
%b-ixcre
*bgon=2
############################################################
######################
# Index Creation Phase
*sql
{
connect tpcd/tpcd;
!date
set echo on
set timing on
rem drop index i_l_orderkey;
create index i_l_orderkey
on lineitem (l_orderkey)
pctfree 5
initrans 10
tablespace ts_i_lorderkey
storage (freelist groups 4 freelists 84)
parallel 144
compute statistics
nologging ;
}
*wait
*sql
{
connect tpcd/tpcd;
!date
set echo on
set timing on
rem drop index i_o_orderkey;
create unique index i_o_orderkey
on orders (o_orderkey)
pctfree 5
initrans 10
tablespace ts_i_oorderkey
storage (freelist groups 4 freelists 84 )
parallel 144
compute statistics
nologging ;
}
*wait
*sql
{
connect tpcd/tpcd;
!date
set echo on
set timing on
rem drop index i_c_custkey;
create unique index i_c_custkey
on customer (c_custkey)
pctfree 0
initrans 10
tablespace ts_i_ccustkey
storage (freelists 84)
parallel 144
compute statistics
nologging ;
}
*wait
*sql
{
connect tpcd/tpcd
set timing on

alter index i_o_orderkey allocate extent (size 10000m
instance 1);
}
*sql
{
connect tpcd/tpcd
set timing on

alter index i_l_orderkey allocate extent (size 20000m
instance 1);
}
*wait
*bgoff
%e-ixcre
%b-anlyz
*bgon=1
############################################################
######################
# Analyze Phase
*sql
{
connect tpcd/tpcd;
!date
set timing on
execute dbms_stats.gather_schema_stats('TPCD' ,
estimate_percent => 1, degree => 144 , granularity =>
'GLOBAL' );
!date
connect / as sysdba
execute dbms_stats.gather_system_stats;
}
*wait
*bgoff
%e-anlyz
```

==========

ACID Test Source Code

==========

==========

atranspl.c

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

#include "atranspl.h"

/* Declare error handling functions */

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

/* declarations for ORDERS */

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

/* declarations for LINEITEM */
```

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

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

sb2 l_npricei;

/* other declarations */

int delta = 0;
double rprice;
double cost;

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

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

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

unsigned flag = (unsigned) 0;       /* flag to store
all sorts of options  */

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

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

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

time_t curr_time;         /* Current Time
*/

/* OCI handles */

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

/* OCI bind handles */

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

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

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

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

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

char sqlstmt[1024];

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

void usage()
{

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

  fprintf(stderr,"    proc_no      :the process number
within this ACID\n");
  fprintf(stderr,"    num_streams  :the total number
of ACID transaction streams\n");
  fprintf(stderr,"    commit       :1 to commit
transaction, abort otherwise\n\n");
  fprintf(stderr,"    delta        :1 to generate new
random delta, otherwise obtain delta from input\n\n");
  fprintf(stderr,"    OPTIONAL PARAMETERS:\n");
  fprintf(stderr,"    i<pathname for input>      :full
path name for input file - default is stdin\n");
  fprintf(stderr,"    o<pathname for output>     :full
path name for output file - default is stdout\n");
  fprintf(stderr,"    d<pathname for durability> :full
path name for durability success file - must specify
for durability test\n");
  fprintf(stderr,"    u<uid/passwd>
:Username/Password string - default is tcpd/tpcd\n");
  fprintf(stderr,"    t<trigger>
:Trigger Time - sleep <trigger> seconds before
start\n\n");
  fprintf(stderr,"    s<sleep>
:Sleep Time - sleep <sleep> seconds before commit or
rollback\n\n");
  exit(-1);

}


void ACIDexit() {

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

}



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

void sql_error(errhp,status,type)
     OCIError *errhp;
```

```
     sword status;
     sword type;
{
  char msg[2048];
  ub4 errcode;
  ub4 msglen;
  int i,j;

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

  (void) OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);

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

  ACIDexit();

  exit(1);
}

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

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

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

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

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

  proc_no = atoi(argv[1]);

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

  num_streams = atoi(argv[2]);

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

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

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

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

  /* Process  optional parameters */

  argc -= 4;
  argv += 4;

  while(--argc) {
    ++argv;
    switch(argv[0][0]) {
    case 'u':
      strncpy((char *) lname, ++(argv[0]), UNAME_LEN);
      if (strchr((char *) lname, '/') == NULL) {
        fprintf(stderr, "Login name must be in the
format of userid/passwd\n");
        usage();
        exit(-1);
      }
      break;
    case 'i':
      if ((infile = fopen(++(argv[0]), "r")) == NULL)
{
        fprintf(stderr,"Cannot open input file %s\n",
argv[0]);
        fprintf(stderr,"%s\n",strerror(errno));
        exit(-1);
      }
      BIS(flag, INFILE);
      break;
    case 'o':
      if ((outfile = open(++(argv[0]), (O_RDWR |
O_SYNC | O_CREAT), S_IRWXU)) == -1) {
        fprintf(stderr,"Cannot open output file %s\n",
argv[0]);
        fprintf(stderr,"%s\n",strerror(errno));
        exit(-1);
      }
      BIS(flag, OUTFILE);
      break;
    case 'd':
      if ((logfile = open(++(argv[0]), (O_RDWR |
O_SYNC | O_CREAT), S_IRWXU)) == -1) {
        fprintf(stderr,"Cannot open durability success
file %s\n", argv[0]);
        fprintf(stderr,"%s\n",strerror(errno));
        exit(-1);
      }
      BIS(flag, LOGFILE);
      break;
    case 'b':
      num_iter = atoi(++(argv[0]));
      break;
    case 't':
      trig = atoi(++(argv[0]));
      break;
    case 's':
      slp = atoi(++(argv[0]));
      break;
    default:
      fprintf(stderr, "Unknown argument %s\n",
argv[0]);
      usage();
      break;
    }
  }

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

  /* Initialize the cursors etc. */

  (void) ACIDinit();
```

```
  /* sleep for some time (triggering) */

  sleep(trig);

  /* start doing the ACID transactions */

  tr_start = gettime();

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

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

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

    /* Obtain l_key from l_key query */

    OCIsexec(tpcsvc,curi,errhp,1);

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

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

    /* Generate delta if necessary */

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

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

    curr_time = time(NULL);

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

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

    OCIsexec(tpcsvc,curr,errhp,1);

    curr_time = time(NULL);

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

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

    /* Shall we commit? */

    if (BIT(flag, COMMIT)) {
      need_commit = 1;
      while (need_commit) {
        if((status=OCITransCommit(tpcsvc,errhp,
OCI_DEFAULT)) != OCI_SUCCESS) {
            OCIrol(tpcsvc,errhp);
            OCIsexec(tpcsvc,curr,errhp,1);
       } else {
           need_commit = 0;
           curr_time = time(NULL);
           FPRTF2(outfile, "ACID Transaction
iteration %d COMMITED at %s\n",
             num_iter, ctime(&curr_time));
         }
       }
     } else {
       OCIrol(tpcsvc,errhp);
       curr_time = time(NULL);
       FPRTF2(outfile, "ACID Transaction iteration %d
ROLLBACK at %s\n",
            num_iter, ctime(&curr_time));
    }

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

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

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

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

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

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

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

  tr_end = gettime();

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

```
  /*  Disconnect from ORACLE. */

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

  ACIDexit();

  exit(0);
}


void ACIDinit()
{

  /* run random seed */

  srand48(getpid());

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

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

  OCIhalloc(tpcenv,&errhp,OCI_HTYPE_ERROR);
  OCIhalloc(tpcenv,&curi,OCI_HTYPE_STMT);
  OCIhalloc(tpcenv,&curr,OCI_HTYPE_STMT);
  OCIhalloc(tpcenv,&cure1,OCI_HTYPE_STMT);
  OCIhalloc(tpcenv,&cure2,OCI_HTYPE_STMT);
  OCIhalloc(tpcenv,&tpcsvc,OCI_HTYPE_SVCCTX);
  OCIhalloc(tpcenv,&tpcsrv,OCI_HTYPE_SERVER);
  OCIhalloc(tpcenv,&tpcusr,OCI_HTYPE_SESSION);

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

  /* get username and password */

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

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

  OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcsrv,0,
OCI_ATTR_SERVER,errhp);
  OCIaset(tpcusr,OCI_HTYPE_SESSION,lname,
strlen(lname),OCI_ATTR_USERNAME,
          errhp);
  OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,
strlen(passwd),OCI_ATTR_PASSWORD,
          errhp);

  if ((status = OCISessionBegin(tpcsvc, errhp, tpcusr,
OCI_CRED_RDBMS,
                                OCI_DEFAULT)) !=
OCI_SUCCESS)
    sql_error(errhp,status,1);

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

  /* Enable session parallel dml */

  sprintf((char *) sqlstmt, PDMLTXT);
  OCIStmtPrepare(curi,errhp,(text *)sqlstmt,
strlen((char *)sqlstmt),
                 OCI_NTV_SYNTAX,OCI_DEFAULT);
  OCIsexec(tpcsvc,curi,errhp,1);

  /* Enable session parallel ddl */

  /*sprintf((char *) sqlstmt, PDDLTXT);
  OCIStmtPrepare(curi,errhp,(text *)sqlstmt,
strlen((char *)sqlstmt),
                 OCI_NTV_SYNTAX,OCI_DEFAULT);
  OCIsexec(tpcsvc,curi,errhp,1);*/


  /* Make session serializable */

  sprintf ((char *) sqlstmt, ISOTXT);
  OCIStmtPrepare(curi,errhp,(text *)sqlstmt,
strlen((char *)sqlstmt),
                 OCI_NTV_SYNTAX,OCI_DEFAULT);
  OCIsexec(tpcsvc,curi,errhp,1);

  /* Set optimizer_index_cost_adj = 25 */

  sprintf ((char *) sqlstmt, OICATXT);
  OCIStmtPrepare(curi,errhp,(text *)sqlstmt,
strlen((char *)sqlstmt),
                 OCI_NTV_SYNTAX,OCI_DEFAULT);
  OCIsexec(tpcsvc,curi,errhp,1);

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


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

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


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

#endif /* NOLKEY */

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

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

  /* bind variables */

  OCIbbname(curr,l_key_bp,errhp,":l_key",ADR(l_key),
SIZ(l_key),SQLT_INT);
  OCIbbname(curr,o_key_bp,errhp,":o_key",ADR(o_key),
SIZ(o_key),SQLT_INT);
  OCIbbname(curr,delta_bp,errhp,":delta",ADR(delta),
SIZ(delta),SQLT_INT);
  OCIbbname(curr,l_pkey_bp,errhp,":l_pkey",
ADR(l_pkey),SIZ(l_pkey),SQLT_INT);
  OCIbbname(curr,l_skey_bp,errhp,":l_skey",
ADR(l_skey),SIZ(l_skey),SQLT_INT);
  OCIbbname(curr,l_quan_bp,errhp,":l_quan",
ADR(l_quan),SIZ(l_quan),SQLT_INT);
  OCIbbname(curr,l_newquan_bp,errhp,":l_newquan",
ADR(l_newquan),
            SIZ(l_newquan),SQLT_INT);
  OCIbbname(curr,l_tax_bp,errhp,":l_tax",ADR(l_tax),
SIZ(l_tax),SQLT_FLT);
  OCIbbname(curr,l_disc_bp,errhp,":l_disc",
ADR(l_disc),SIZ(l_disc),SQLT_FLT);
  OCIbbname(curr,l_eprice_bp,errhp,":l_eprice",
ADR(l_eprice),SIZ(l_eprice),
            SQLT_FLT);
  OCIbbname(curr,l_neweprice_bp,errhp,":l_neweprice",
ADR(l_neweprice),
            SIZ(l_neweprice),SQLT_FLT);

  OCIbbname(curr,o_tprice_bp,errhp,":o_tprice",
```

```
ADR(o_tprice),SIZ(o_tprice),
            SQLT_FLT);
  OCIbbname(curr,o_newtprice_bp,errhp,":o_newtprice",
ADR(o_newtprice),
            SIZ(o_newtprice), SQLT_FLT);
  OCIbbname(curr,rprice_bp,errhp,":rprice",
ADR(rprice),SIZ(rprice), SQLT_FLT);
  OCIbbname(curr,cost_bp,errhp,":cost",ADR(cost),
SIZ(cost), SQLT_FLT);

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

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

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

  /* bind variables */

  OCIbbname(cure1,l_neweprice1_bp,
errhp,":l_neweprice",ADR(l_neweprice),
            SIZ(l_neweprice),SQLT_FLT);
  OCIbbname(cure1,l_newquan1_bp,errhp,":l_newquan",
ADR(l_newquan),
            SIZ(l_newquan),SQLT_INT);
  OCIbbname(cure1,o_key1_bp,errhp,":o_key",ADR(o_key),
SIZ(o_key),SQLT_INT);
  OCIbbname(cure1,l_key1_bp,errhp,":l_key",ADR(l_key),
SIZ(l_key),SQLT_INT);

  OCIbbname(cure2,o_newtprice2_bp,
errhp,":o_newtprice",ADR(o_newtprice),
            SIZ(o_newtprice),SQLT_FLT);
  OCIbbname(cure2,o_key2_bp,errhp,":o_key",ADR(o_key),
SIZ(o_key),SQLT_INT);

}
```

==========

atranspl.h

```
#ifndef ATRANSPL_H

#define ATRANSPL_H

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

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

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

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

extern int errno;

#ifndef NULL
#define NULL 0
#endif

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

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#ifndef ub1
#define ub1 unsigned char
#endif

#define UNAME_LEN 64
#define WRITE_BUF_LEN 1024

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

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

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

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

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

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

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

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

#define OCIbbname(stmh,bindp,errh,sqlvar,progv,progvl,
ftype) \
    if((status=OCIBindByName(stmh,&bindp,errh,(text
*)sqlvar,strlen(sqlvar), \
                progv,progvl,ftype,0,0,0,0,0,
OCI_DEFAULT)) != OCI_SUCCESS) \
       sql_error(errh,status,1); \
    else \
       DISCARD 0

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

```
    if((status=OCIBindByName(stmh,&bindp,errh,(text
*)sqlvar,strlen(sqlvar), \
                progv,progvl,ftype,indp,0,0,0,0,
OCI_DEFAULT)) != OCI_SUCCESS) \
       sql_error(errh,status,1); \
    else \
       DISCARD 0

#define OCIcom(svcp,errh) \
     if((status=OCITransCommit(svcp,errh,OCI_DEFAULT))
!= OCI_SUCCESS) \
        sql_error(errh,status,1); \
     else \
        DISCARD 0

#define OCIrol(svcp,errh) \
     if((status=OCITransRollback(svcp,errh,
OCI_DEFAULT)) != OCI_SUCCESS) \
        sql_error(errh,status,1); \
     else \
        DISCARD 0

#define ISOTXT "alter session set isolation_level =
serializable"
#define PDMLTXT "alter session force parallel dml
parallel (degree 84)"
#define PDDLTXT "alter session force parallel ddl
parallel (degree 84)"
#define OICATXT "alter session set
optimizer_index_cost_adj=25"

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

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

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

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

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

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

#endif /* ATRANSPL_H */
```

==========

randpsup.c

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

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

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

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

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


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

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

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

  /* seed the random number generator */

  srand48(getpid());

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

  PART_SUPP_BRIDGE(ps_skey, pkey, supp);

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

  exit(0);
}


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

==========

randkey.c

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

#define ORDERCNT 150000.0

/* MK_SPARSE adopted from dss.h */

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

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

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

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

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

```
/* OCI handles */

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

OCIBind *l_key_bp;
OCIBind *o_key_bp;

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

char sqlstmt[1024];


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


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

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

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

  (void) OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);

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

  ACIDexit();

  exit(1);
}


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

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

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

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

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

  argc -= 2;
  argv += 2;

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

  ACIDinit();

  /* initialize array for random numbers */

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

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

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

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

    /* Obtain l_key from l_key query */

    OCIsexec(tpcsvc,curi,errhp,1);

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

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

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

```
  ACIDexit();
  free(res);

}


void usage() {

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

void ACIDinit()
{

  /* run random seed */

  srand48(getpid());

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

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

  OCIhalloc(tpcenv,&errhp,OCI_HTYPE_ERROR);
  OCIhalloc(tpcenv,&curi,OCI_HTYPE_STMT);
  OCIhalloc(tpcenv,&tpcsvc,OCI_HTYPE_SVCCTX);
  OCIhalloc(tpcenv,&tpcsrv,OCI_HTYPE_SERVER);
  OCIhalloc(tpcenv,&tpcusr,OCI_HTYPE_SESSION);

  /* get username and password */

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

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

  OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcsrv,0,
OCI_ATTR_SERVER,errhp);
  OCIaset(tpcusr,OCI_HTYPE_SESSION,lname,
strlen(lname),OCI_ATTR_USERNAME,
          errhp);
  OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,
strlen(passwd),OCI_ATTR_PASSWORD,
          errhp);

  if ((status = OCISessionBegin(tpcsvc, errhp, tpcusr,
OCI_CRED_RDBMS,
                                OCI_DEFAULT)) !=
OCI_SUCCESS)
    sql_error(errhp,status,1);

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

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

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

  OCIbbname(curi,l_key_bp,errhp,":l_key",ADR(l_key),
SIZ(l_key),SQLT_INT);
  OCIbbname(curi,o_key_bp,errhp,":o_key",ADR(o_key),
SIZ(o_key),SQLT_INT);
}
```

==========

gettime.c

```
#define SUN_OS5

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

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

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

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

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

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

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

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

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

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

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

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

# include <stdio.h>

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

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


double gettime ()

{

#ifdef TIME_W_GETTIME
```

```
   struct timeval tv;

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

#ifdef TIME_W_TIMES
   struct tms buf;

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

}



double getcpu ()

{

#ifdef CPU_W_TIMES
   struct tms buf;

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

#ifdef CPU_W_GETRU
   struct rusage ru;
   double usecs;

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

}



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

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

{

#ifdef GETRU_STATS
   struct rusage ru;

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

#ifdef GET_P_STATS
   timeval_t tv;
   struct process_stats ru;

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

}



getru1 (kids)

int kids;

{

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

#ifdef GET_P_STATS
   timeval_t tv;

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

}



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

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

{

#ifdef GETRU_STATS
   struct rusage ru;

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

#ifdef GET_P_STATS
   timeval_t tv;
   struct process_stats ru;

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

}



#ifdef GETRU_STATS
```

```
print_ru (fp, ru)

FILE *fp;
struct rusage *ru;

{

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

}



diffru (ru2, ru)

struct rusage *ru2;
struct rusage *ru;

{

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

}

#endif /* GETRU_STATS */



#ifdef GET_P_STATS

print_ru (fp, ps)

FILE *fp;
struct process_stats *ps;

{

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

}



diffru (ru2, ru)

struct process_stats *ru2;
struct process_stats *ru;

{

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

}

#endif /* GET_P_STATS */
```

==========

a_query.sql

==========

```
set serverout on;

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

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

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

exit;
```

==========

a_query2.sql

==========

```
set serverout on;

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

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

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

exit;
```

==========

atom.sh

==========

```
#!/bin/ksh

. $KIT_DIR/env

OH=$ORACLE_HOME
# ACID_DIR=$TPCD_KIT_DIR/audit set in env
OUT_DIR=$ACID_OUT
DURA_DIR=$ACID_DIR/dura
SF=1

usage() {

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

ITER=3
SF=1
PROG=atranspl
OUT=${OUT_DIR}/atom
USER=tpcd/tpcd

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

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

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

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

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

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

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

==========

atranspl.sh

==========

```
#!/bin/ksh
ade useview acid2 << END
echo I am in atranspl.sh
echo my arguments are $1 $2 $3 $4 $5 $6 $7
/private/tpcd/acid/kit/utils/atranspl $1 $2 $3 $4 $5
$6 $7
exit
END
```

==========

ckpt.sh

==========

```
#!/bin/ksh

. $KIT_DIR/env

svrmgrl << !

       connect internal;
       alter system switch logfile;
       alter system switch logfile;
       exit;
!
```

==========

cnt_hist.sql

==========

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

==========

consist.sh

==========

```
#!/bin/ksh

. $KIT_DIR/env

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

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

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

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

STREAM=9
ITER=100
PROG=atranspl
USER="tpcd/tpcd"
CK=10

usage() {

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

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

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


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

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

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

i=0

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

echo "Check consistency before Submitting Transactions
`date`"
echo "Check consistency before Submitting Transactions
`date`" >> $CON1
echo "Obtain 10 keys from the each key file to check
consistency"

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

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

i=0

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

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

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

wait

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

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

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

cat ${ORACLE_HOME}/rdbms/log/alert_${ORACLE_SID}.log
>> $CHK

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

==========

consist.sql

==========

```
set verify off

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

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


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

    diff := l_tprice - o_tprice;

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

END;
.
/


spool off
exit
```

==========

end_acid.sh

==========

```
#!/bin/ksh

. $KIT_DIR/env

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

ITER=1000
STEM=9
PROG=${ACID_DIR}/atranspl.ott
IN=${ACID_DIR}/acid_in
DURA=${DURA_DIR}/dura
DURA=${DURA_DIR}/drate
OUT=${DURA_DIR}/drate
DSMPL=${DURA_DIR}/durasmpl
KEY=${DURA_DIR}/key${1}_
USER=tpcd/tpcd
TRIG=1
HCNT=duracnta

sqlplus $USER @cnt_hist > $DURA_DIR/$HCNT

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

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

==========

iso1.sh

==========

```
#!/bin/ksh

. $KIT_DIR/env

RSH=rsh

OH=$ORACLE_HOME
OUT_DIR=$ACID_OUT

TXN1FILE=$OUT_DIR/txn1$$.out
TXN2FILE=$OUT_DIR/txn2$$.out
KEYFILE=$OUT_DIR/key$$.out
ISOFILE=$OUT_DIR/iso1

USER="tpcd/tpcd"
PROG=atranspl


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

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

usage() {

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

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

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


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

randkey 1 1 u"$USER" > $KEYFILE
echo -----
ls -l $KEYFILE
cat $KEYFILE
echo ++++++
OKEY=`cat $KEYFILE | awk '{print $1}'`
echo "o_key is "$OKEY

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

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

echo PROG 1 1 1 0 i$KEYFILE u$USER s60

sleep 10

echo "Running ACID query 10 seconds AFTER the start of
ACID Transaction" \
>> $TXN2FILE
echo "`date`" >> $TXN2FILE
if [ "$HOST" != "" ]
then
echo "Starting ACID query on node $HOST" >> $TXN2FILE
@$ACID_DIR/isolation/a_query $OKEY >> $TXN2FILE
${RSH} -n ${HOST} a_query.sh $OKEY >> $TXN2FILE
else
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >>
$TXN2FILE
fi

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

==========

iso2.sh

==========

```
#!/bin/ksh

. $KIT_DIR/env

RSH=rsh

OH=$ORACLE_HOME
OUT_DIR=$ACID_OUT

DURA_DIR=$ACID_DIR/dura

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

USER="tpcd/tpcd"
PROG=atranspl

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

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

usage() {

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

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

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


randkey 1 1 u"$USER" > $KEYFILE

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

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

sleep 1

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

sleep 10

echo "Running ACID query 10 seconds AFTER the start of
ACID transaction" \
>> $TXN2FILE
echo "`date`" >> $TXN2FILE
if [ "$HOST" != "" ]
then
echo "Starting ACID query on node $HOST" >> $TXN2FILE
${RSH} -n ${HOST} a_query.sh $OKEY >> $TXN2FILE
else
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >>
$TXN2FILE
fi

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

=====================================================
```

iso3.sh

```
=====================================================
#!/bin/ksh
. $KIT_DIR/env

RSH=rsh

OH=$ORACLE_HOME
OUT_DIR=$ACID_OUT

DURA_DIR=$ACID_DIR/dura

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

USER="tpcd/tpcd"
PROG=$KIT_DIR/utils/atranspl

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

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

usage() {

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

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

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

randkey 1 1 u"$USER" > $KEYFILE
k1=`cat $KEYFILE | awk '{print $1}'`
k2=`cat $KEYFILE | awk '{print $2}'`
k3=`cat $KEYFILE | awk '{print $3}'`
sleep 1

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

sleep 10

if [ "$HOST" != "" ]
then
echo "Starting TXN2 on node $HOST" >> $TXN2FILE
${RSH} -n ${HOST} atranspl.sh 2 2 1 1 $k1 $k2 $k3
u$USER s1 >> $TXN2FILE &
else
$PROG 2 2 1 1 i$KEYFILE u$USER s1 >> $TXN2FILE &
fi

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

=====================================================
```

iso4.sh

```
=====================================================
#!/bin/ksh
. $KIT_DIR/env

RSH=rsh

OH=$ORACLE_HOME
OUT_DIR=$ACID_OUT

DURA_DIR=$ACID_DIR/dura

TXN1FILE=$OUT_DIR/txn1$$.out
TXN2FILE=$OUT_DIR/txn2$$.out
KEYFILE=/tpch_ff/ff_ps8/key$$.out
ISOFILE=$OUT_DIR/iso4

USER="tpcd/tpcd"
PROG=atranspl

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

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

usage() {

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

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

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

randkey 1 1 u"$USER" > $KEYFILE

k1=`cat $KEYFILE | awk '{print $1}'`
k2=`cat $KEYFILE | awk '{print $2}'`
k3=`cat $KEYFILE | awk '{print $3}'`


sleep 1

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

sleep 10

if [ "$HOST" != "" ]
then
echo "Starting TXN2 on node $HOST" >> $TXN2FILE
${RSH} -n ${HOST} atranspl.sh 2 2 1 1 $k1 $k2 $k3
u$USER s1 >> $TXN2FILE &
else
$PROG 2 2 1 1 i$KEYFILE u$USER s1 >> $TXN2FILE &
fi

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

==========

iso5.sh

==========

```
#!/bin/ksh
. $KIT_DIR/env

RSH=rsh

OH=$ORACLE_HOME
OUT_DIR=$ACID_OUT
DURA_DIR=$ACID_DIR/dura

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

USER="tpcd/tpcd"
PROG=atranspl

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

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

usage() {

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

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

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

randkey 1 1 u"$USER" > $KEYFILE

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

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

sleep 1

$PROG 1 1 1 0 i$KEYFILE u$USER s60 >> $TXN1FILE &
sleep 5
PSKEY=`randpsup 1`

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

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

wait

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

==========

iso6.sh

==========

```
#!/bin/ksh
. $KIT_DIR/env

# May need to change the following:
RSH=rsh

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

DURA_DIR=$ACID_DIR/dura
```

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

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

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

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

sleep 1

# start Query 1, use 0 as the delta

echo "Running Query 1 at `date`" >> $TXN1FILE
sqlplus $USER @q1 >> $TXN1FILE &

# sleep 2 seconds before starting ACID transaction

sleep 2

# start ACID transaction, COMMIT after one second

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

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

# start Query 1

sleep 2

echo "Running 2nd Query 1 at `date`" >> $TXN3FILE
sqlplus $USER @q1 >> $TXN3FILE &
# wait for everyone to finish

wait

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

cat $TXN1FILE $TXN2FILE $TXN3FILE >> $ISOFILE

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

==========

run_acid.sh

==========

```
#!/bin/ksh

. $KIT_DIR/env

OH=$ORACLE_HOME
OUT_DIR=$ACID_OUT

usage() {

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

ITER=1000
STEM=9
SF=1
PROG=atranspl
IN=${ACID_DIR}/acid_in
DURA_DIR=$ACID_OUT/dura
OUT=${DURA_DIR}/drate
DURA=${DURA_DIR}/dura
KEY=${DURA_DIR}/key$$_
USER=tpcd/tpcd
TRIG=1
HCNT=duracntb

set -- `getopt "n:s:p:i:o:d:u:ht:f:" "$@"` || usage
while :
do
   case "$1" in
   -n) shift; ITER=$1;;
   -s) shift; STEM=$1;;
   -p) shift; PROG=$1;;
   -i) shift; IN=$1;;
   -o) shift; OUT=$1;;
   -d) shift; DURA=$1;;
   -u) shift; USER=$1;;
   -h) usage; exit 0;;
```

```
   -t) shift; TRIG=$1;;
   -f) shift; SF=$1;;
   --) break;;
   esac
   shift;
done


echo "Starting durability run..."

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

sqlplus -s $USER << ! > $DURA_DIR/$HCNT 2>&1
connect tpcd/tpcd
@cnt_hist
!

while [ $i -lt $STEM ]
do
   randkey 1000 ${SF} u${USER} > ${KEY}${i} &
   i=`expr $i + 1`
done

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

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

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

done

wait

echo "Durability run completed"
```

==========

sample.sh

==========

```
#!/bin/ksh
. $KIT_DIR/env

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

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

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

i=1
while [ $i -le 6 ]
do

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

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

==========

sample.sql

==========

```
alter session set nls_date_format = 'YYYY-MM-DD
HH:MI:SS';

select * from history where  h_o_key = &&1 and h_l_key
= &&2;

exit;
```

==========

Disk Configuration Details

==========

Basic Assumptions:

1. Use T3 with write cache for redo logs.
2. Use A5200 striped wide for everything else.

Disk Config:

2 - T3 bricks 9 x 18G drives.
1 - A5200 2 x 9GB /* boot array */
33 - A5200 12 10Krpm 73.4G drives /* data and indexes/

Redo Logs:

- Create 9-disk RAID0 LUN on each T3 brick (64K blocksize).
- Turn write cache on for each LUN.
- Using Veritas, Mirror between the two luns.

Data/indexes:

Goals:

1. Stripe every logical entity (table, index, temp) to as many
    of the 396 disks as possible.

2. Keep the number of objects in a disk group
   manageable.

Limitations:

1. Striping volumes 396 wide would limit the number of volumes to be created.

What we did was to create 6 VxVM disk groups on the A5200 rotating through the trays so each disk group got approximately the same number of disks from each tray, with 66 disks per disk group.

For RAID1+0 (stripe-mirrors) volumes (all but temp), We created 66 mirrors from each controller and striped them into a single volume. Every logical entity was created on multiple volumes, e.g. the tablespace for LINEITEM was created on 84 volumes. Using round-robin assignment, these volumes rotated between the 6 disk groups. In this way the LINEITEM table was spread across all 396 disks.

For RAID0 volumes (temp), we striped across all 66 disks from each disk group. Again the logical entity (e.g. tablespace for temp) was created on multiple volumes so that again it ended up striped across all 396 disks.

Each drive was accessible via two controllers using the Veritas DMP driver. This allows for load balancing and failover capabilities.

Following is the volume manager description of typical volumes for each entity.

```
V  NAME          RVG           KSTATE   STATE    LENGTH
READPOL   PREFPLEX UTYPE
PL NAME        VOLUME        KSTATE   STATE    LENGTH   LAYOUT
NCOL/WID MODE
SD NAME        PLEX          DISK     DISKOFFS LENGTH   [COL/]OFF DEVICE
MODE
SV NAME        PLEX          VOLNAME  NVOLLAYR LENGTH   [COL/]OFF AM/NM
MODE
DC NAME        PARENTVOL     LOGVOL
SP NAME        SNAPVOL       DCO

v  l_1         -             ENABLED  ACTIVE   82046976 PREFER    -
gen
pl l_1-03      l_1           ENABLED  ACTIVE   82046976 STRIPE
66/2048  RW
sv l_1-S01     l_1-03        l_1-L01  1        1243136  0/0       2/2
```

```
ENA
sv l_1-S02      l_1-03      l_1-L02  1       1243136  1/0      2/2
ENA
sv l_1-S03      l_1-03      l_1-L03  1       1243136  2/0      2/2
ENA
sv l_1-S04      l_1-03      l_1-L04  1       1243136  3/0      2/2
ENA
sv l_1-S05      l_1-03      l_1-L05  1       1243136  4/0      2/2
ENA
sv l_1-S06      l_1-03      l_1-L06  1       1243136  5/0      2/2
ENA
sv l_1-S07      l_1-03      l_1-L07  1       1243136  6/0      2/2
ENA
sv l_1-S08      l_1-03      l_1-L08  1       1243136  7/0      2/2
ENA
sv l_1-S09      l_1-03      l_1-L09  1       1243136  8/0      2/2
ENA
sv l_1-S10      l_1-03      l_1-L10  1       1243136  9/0      2/2
ENA
sv l_1-S11      l_1-03      l_1-L11  1       1243136  10/0     2/2
ENA
sv l_1-S12      l_1-03      l_1-L12  1       1243136  11/0     2/2
ENA
sv l_1-S13      l_1-03      l_1-L13  1       1243136  12/0     2/2
ENA
sv l_1-S14      l_1-03      l_1-L14  1       1243136  13/0     2/2
ENA
sv l_1-S15      l_1-03      l_1-L15  1       1243136  14/0     2/2
ENA
sv l_1-S16      l_1-03      l_1-L16  1       1243136  15/0     2/2
ENA
sv l_1-S17      l_1-03      l_1-L17  1       1243136  16/0     2/2
ENA
sv l_1-S18      l_1-03      l_1-L18  1       1243136  17/0     2/2
ENA
sv l_1-S19      l_1-03      l_1-L19  1       1243136  18/0     2/2
ENA
sv l_1-S20      l_1-03      l_1-L20  1       1243136  19/0     2/2
ENA
sv l_1-S21      l_1-03      l_1-L21  1       1243136  20/0     2/2
ENA
sv l_1-S22      l_1-03      l_1-L22  1       1243136  21/0     2/2
ENA
sv l_1-S23      l_1-03      l_1-L23  1       1243136  22/0     2/2
ENA
sv l_1-S24      l_1-03      l_1-L24  1       1243136  23/0     2/2
ENA
sv l_1-S25      l_1-03      l_1-L25  1       1243136  24/0     2/2
ENA
sv l_1-S26      l_1-03      l_1-L26  1       1243136  25/0     2/2
ENA
sv l_1-S27      l_1-03      l_1-L27  1       1243136  26/0     2/2
ENA
sv l_1-S28      l_1-03      l_1-L28  1       1243136  27/0     2/2
ENA
sv l_1-S29      l_1-03      l_1-L29  1       1243136  28/0     2/2
ENA
sv l_1-S30      l_1-03      l_1-L30  1       1243136  29/0     2/2
ENA
sv l_1-S31      l_1-03      l_1-L31  1       1243136  30/0     2/2
ENA
sv l_1-S32      l_1-03      l_1-L32  1       1243136  31/0     2/2
ENA
sv l_1-S33      l_1-03      l_1-L33  1       1243136  32/0     2/2
ENA
sv l_1-S34      l_1-03      l_1-L34  1       1243136  33/0     2/2
ENA
sv l_1-S35      l_1-03      l_1-L35  1       1243136  34/0     2/2
ENA
sv l_1-S36      l_1-03      l_1-L36  1       1243136  35/0     2/2
ENA
sv l_1-S37      l_1-03      l_1-L37  1       1243136  36/0     2/2
ENA
sv l_1-S38      l_1-03      l_1-L38  1       1243136  37/0     2/2
ENA
sv l_1-S39      l_1-03      l_1-L39  1       1243136  38/0     2/2
ENA
sv l_1-S40      l_1-03      l_1-L40  1       1243136  39/0     2/2
ENA
sv l_1-S41      l_1-03      l_1-L41  1       1243136  40/0     2/2
ENA
sv l_1-S42      l_1-03      l_1-L42  1       1243136  41/0     2/2
ENA
sv l_1-S43      l_1-03      l_1-L43  1       1243136  42/0     2/2
ENA
sv l_1-S44      l_1-03      l_1-L44  1       1243136  43/0     2/2
ENA
sv l_1-S45      l_1-03      l_1-L45  1       1243136  44/0     2/2
ENA
sv l_1-S46      l_1-03      l_1-L46  1       1243136  45/0     2/2
ENA
sv l_1-S47      l_1-03      l_1-L47  1       1243136  46/0     2/2
ENA
sv l_1-S48      l_1-03      l_1-L48  1       1243136  47/0     2/2
ENA
sv l_1-S49      l_1-03      l_1-L49  1       1243136  48/0     2/2
ENA
sv l_1-S50      l_1-03      l_1-L50  1       1243136  49/0     2/2
ENA
sv l_1-S51      l_1-03      l_1-L51  1       1243136  50/0     2/2
ENA
sv l_1-S52      l_1-03      l_1-L52  1       1243136  51/0     2/2
ENA
sv l_1-S53      l_1-03      l_1-L53  1       1243136  52/0     2/2
ENA
sv l_1-S54      l_1-03      l_1-L54  1       1243136  53/0     2/2
ENA
sv l_1-S55      l_1-03      l_1-L55  1       1243136  54/0     2/2
ENA
sv l_1-S56      l_1-03      l_1-L56  1       1243136  55/0     2/2
ENA
sv l_1-S57      l_1-03      l_1-L57  1       1243136  56/0     2/2
ENA
sv l_1-S58      l_1-03      l_1-L58  1       1243136  57/0     2/2
ENA
sv l_1-S59      l_1-03      l_1-L59  1       1243136  58/0     2/2
ENA
sv l_1-S60      l_1-03      l_1-L60  1       1243136  59/0     2/2
ENA
sv l_1-S61      l_1-03      l_1-L61  1       1243136  60/0     2/2
ENA
sv l_1-S62      l_1-03      l_1-L62  1       1243136  61/0     2/2
ENA
sv l_1-S63      l_1-03      l_1-L63  1       1243136  62/0     2/2
ENA
sv l_1-S64      l_1-03      l_1-L64  1       1243136  63/0     2/2
ENA
sv l_1-S65      l_1-03      l_1-L65  1       1243136  64/0     2/2
ENA
sv l_1-S66      l_1-03      l_1-L66  1       1243136  65/0     2/2
ENA
```

Appendix C. Query Text and Query Output

```
==================================================
```

qual01.v1

```
==================================================
Begin Execution at Fri Apr  4 11:52:01 2003


-- using default substitutions

select
l_returnflag,
l_linestatus,
sum(l_quantity) as sum_qty,
sum(l_extendedprice) as sum_base_price,
sum(l_extendedprice * (1 - l_discount)) as
sum_disc_price,
sum(l_extendedprice * (1 - l_discount) * (1 + l_tax))
as sum_charge,
avg(l_quantity) as avg_qty,
avg(l_extendedprice) as avg_price,
avg(l_discount) as avg_disc,
count(*) as count_order
from
lineitem
where
l_shipdate <= to_date ('1998-12-01','YYYY-MM-DD')  -
90
group by
l_returnflag,
l_linestatus
order by
l_returnflag,
l_linestatus

L_RETURNFLAG L_LINESTATUS SUM_QTY
SUM_BASE_PRICE
SUM_DISC_PRICE         SUM_CHARGE             AVG_QTY
AVG_PRICE              AVG_DISC
COUNT_ORDER
A            F            37734107.00
56586554400.73
53758257134.87         55909065222.83         25.52
38273.13               0.05
1478493.00
N            F              991417.00
1487504710.38
1413082168.05          1469649223.19          25.52
38284.47               0.05                   38854.00
N            O            74476040.00
111701729697.74
106118230307.61        110367043872.50        25.50
38249.12               0.05
2920374.00
R            F            37719753.00
56568041380.90
53741292684.60         55889619119.83         25.51
38250.85               0.05
1478870.00


4 rows processed.
Statement Processed in 13.48 seconds.

Ended Executing this Stream at Fri Apr  4 11:52:15
2003


Stream Started at 1049485921.62
Stream Ended at 1049485935.10
Stream Processed in 13.48 seconds


SQL statements processed: 1
==================================================
```

qual02.v1

```
==================================================
Begin Execution at Fri Apr  4 11:52:15 2003


-- using default substitutions

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

S_ACCTBAL              S_NAME
N_NAME
P_PARTKEY              P_MFGR
S_ADDRESS                                S_PHONE
S_COMMENT
9938.53                Supplier#000005359
UNITED KINGDOM
185358.00              Manufacturer#4
bgxj2K0w1kJvxYl5mhCfou,W
33-429-790-6131
blithely silent pinto beans are furiously. slyly final
deposits acros
9937.84                Supplier#000005969
ROMANIA
108438.00              Manufacturer#1
rdnmd9c8EG1EIAYY3LPVa4yUNx6OwyVaQ
29-520-692-3537
carefully slow deposits use furiously. slyly ironic
platelets above the ironic
9936.22                Supplier#000005250
UNITED KINGDOM
249.00                 Manufacturer#4
qX AB0vP8mJEWeBuY9jri
33-320-228-2957
blithely special packages are. stealthily express
deposits across the closely final instructi
9923.77                Supplier#000002324
GERMANY
29821.00               Manufacturer#4
uXcnR7tv87dG
17-779-299-1839
quickly express packages breach quiet pinto beans.
requ
9871.22                Supplier#000006373
GERMANY
43868.00               Manufacturer#5
iSLO35z7Ae
17-813-485-8637
never silent deposits integrate furiously blit
9870.78                Supplier#000001286
```

```
GERMANY
81285.00                Manufacturer#2
3gq0mZLHI5OTM6 tBYmLTHZaulCYnlECzQ7nj
17-516-924-4574
final theodolites cajole slyly special,
9870.78                 Supplier#000001286
GERMANY
181285.00               Manufacturer#4
3gq0mZLHI5OTM6 tBYmLTHZaulCYnlECzQ7nj
17-516-924-4574
final theodolites cajole slyly special,
9852.52                 Supplier#000008973
RUSSIA
18972.00                Manufacturer#2
zVfUT3Np22kUC05tYWHBotaR
32-188-594-7038
quickly regular instructions wake-- carefully unusual
braids into the expres
9847.83                 Supplier#000008097
RUSSIA
130557.00               Manufacturer#2
veMRTQBmUResNvfD3
32-375-640-3593
slyly regular dependencies sleep slyly furiously
express dep
9847.57                 Supplier#000006345
FRANCE
86344.00                Manufacturer#1
68yX tGXAkVRSxUGNSjJdptw 8O878xaFnaoQK
16-886-766-7945
silent pinto beans should have to snooze carefully
along the final reques
9847.57                 Supplier#000006345
FRANCE
173827.00               Manufacturer#2
68yX tGXAkVRSxUGNSjJdptw 8O878xaFnaoQK
16-886-766-7945
silent pinto beans should have to snooze carefully
along the final reques
9836.93                 Supplier#000007342
RUSSIA
4841.00                 Manufacturer#4
icFgTpZ0TuAm188dv
32-399-414-5385
final accounts haggle. bold accounts are furiously
dugouts. furiously silent asymptotes are slyly
9817.10                 Supplier#000002352
RUSSIA
124815.00               Manufacturer#2
XfLCj71HKHnPqgvs7KNgPKcOWoWxo2w
32-551-831-1437
blithely pending packages across the ironic accounts
grow slyly after the furiously
9817.10                 Supplier#000002352
RUSSIA
152351.00               Manufacturer#3
XfLCj71HKHnPqgvs7KNgPKcOWoWxo2w
32-551-831-1437
blithely pending packages across the ironic accounts
grow slyly after the furiously
9739.86                 Supplier#000003384
FRANCE
138357.00               Manufacturer#2
D01XwXbcILNwmrGS6ZPrVhZxO40i
16-494-913-5925
slyly ironic theodolites hag
9721.95                 Supplier#000008757
... rows truncated ....
FRANCE
118574.00               Manufacturer#1
TyptNE7nv6BLpLl6WFX
16-974-998-8937
regular pinto beans are after
7980.65                 Supplier#000001288
FRANCE
13784.00                Manufacturer#4
tm0TjL5b oE
16-646-464-8247
unusual pinto beans cajole furiously according t
7950.37                 Supplier#000008101
GERMANY
33094.00                Manufacturer#5
HG2wfVixwCIhK7dlrigGR3an2LuSifDJH
17-627-663-8014
quickly regular requests are furiously. pending
deposits wake
7937.93                 Supplier#000009012
ROMANIA
83995.00                Manufacturer#2
J6I7sJj0mGYIWFv9KxD3fK O7tvNP
29-250-925-9690
blithely bold ideas haggle quickly final, regular
request
7914.45                 Supplier#000001013
RUSSIA
125988.00               Manufacturer#2
AI9ODzBzWgnny28PHBei5M2lUFdD9
32-194-698-3365
final, ironic theodolites alongside of the ironic
7912.91                 Supplier#000004211
GERMANY
159180.00               Manufacturer#5
Zva95Dwj EY0w,XjgsL7O0Zb2 l3almck
17-266-947-7315
final requests integrate slyly above the silent, even
7912.91                 Supplier#000004211
GERMANY
184210.00               Manufacturer#4
Zva95Dwj EY0w,XjgsL7O0Zb2 l3almck
17-266-947-7315
final requests integrate slyly above the silent, even
7894.56                 Supplier#000007981
GERMANY
85472.00                Manufacturer#4
e8hRUxe9cqQM3b
17-963-404-3760
regular, even theodolites integrate carefully. bold,
special theodolites are slyly fluffily iron
7887.08                 Supplier#000009792
GERMANY
164759.00               Manufacturer#3
3YSi76M2 I8XGikO5YgSM81r5Z6A7VkZcys
17-988-938-4296
pending, ironic packages sleep among the carefully
ironic accounts. quickly final accounts
7871.50                 Supplier#000007206
RUSSIA
104695.00               Manufacturer#1
YvrLdpD 5ExhHmRWzK41tw4
32-432-452-7731
furiously dogged pinto beans cajole. bold, express
notornis until the slyly pending
7852.45                 Supplier#000005864
RUSSIA
8363.00                 Manufacturer#4
5odLpc1M83KXJ0O
32-454-883-3821
blithely regular deposits
7850.66                 Supplier#000001518
UNITED KINGDOM
86501.00                Manufacturer#1
ddNQX3hIjgico
33-730-383-3892
furiously final accounts wake carefully idle requests.
even dolphins wake acc
7843.52                 Supplier#000006683
FRANCE
11680.00                Manufacturer#4
Z1,hkHIw,Z3,,Comv6kLxIiPJtoNt
16-464-517-8943
carefully bold accounts doub


100 rows processed.
Statement Processed in 3.59 seconds.

Ended Executing this Stream at Fri Apr  4 11:52:18
2003


Stream Started at 1049485935.26
Stream Ended at 1049485938.85
Stream Processed in 3.59 seconds


SQL statements processed: 1

==================================================
```

qual03.v1

```
==================================================
```

```
Begin Execution at Fri Apr  4 11:52:19 2003


-- using default substitutions

select * from (
select
l_orderkey,
sum(l_extendedprice * (1 - l_discount)) as revenue,
o_orderdate,
o_shippriority
from
customer,
orders,
lineitem
where
c_mktsegment = 'BUILDING'
and c_custkey = o_custkey
and l_orderkey = o_orderkey
and o_orderdate < to_date( '1995-03-15', 'YYYY-MM-DD')
and l_shipdate > to_date( '1995-03-15', 'YYYY-MM-DD')
group by
l_orderkey,
o_orderdate,
o_shippriority
order by
revenue desc,
o_orderdate)
where rownum <= 10

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


10 rows processed.
Statement Processed in 13.37 seconds.

Ended Executing this Stream at Fri Apr  4 11:52:32
2003


Stream Started at 1049485939.01
Stream Ended at 1049485952.38
Stream Processed in 13.37 seconds


SQL statements processed: 1
```

==========

qual04.v1

```
==========
Begin Execution at Fri Apr  4 11:52:32 2003


-- using default substitutions

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

O_ORDERPRIORITY ORDER_COUNT
1-URGENT        10594.00
2-HIGH          10476.00
3-MEDIUM        10410.00
4-NOT SPECIFIED 10556.00
5-LOW           10487.00


5 rows processed.
Statement Processed in 11.89 seconds.

Ended Executing this Stream at Fri Apr  4 11:52:44
2003


Stream Started at 1049485952.55
Stream Ended at 1049485964.44
Stream Processed in 11.89 seconds


SQL statements processed: 1
```

==========

qual05.v1

```
==========
Begin Execution at Fri Apr  4 11:52:44 2003


-- using default substitutions

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

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


5 rows processed.
Statement Processed in 14.80 seconds.

Ended Executing this Stream at Fri Apr  4 11:52:59
2003
```

```
Stream Started at 1049485964.60
Stream Ended at 1049485979.40
Stream Processed in 14.80 seconds


SQL statements processed: 1
```

==========

qual06.v1

==========

```
Begin Execution at Fri Apr  4 11:52:59 2003


-- using default substitutions

select
sum(l_extendedprice * l_discount) as revenue
from
lineitem
where
l_shipdate >= to_date( '1994-01-01',  'YYYY-MM-DD')
and l_shipdate < add_months(to_date( '1994-01-01',  'YYYY-MM-DD'),  12)
and l_discount between .06 - 0.01 and .06 + 0.01
and l_quantity < 24

REVENUE
123141078.23


1 row# processed.
Statement Processed in 1.74 seconds.

Ended Executing this Stream at Fri Apr  4 11:53:01 2003


Stream Started at 1049485979.55
Stream Ended at 1049485981.30
Stream Processed in 1.75 seconds


SQL statements processed: 1
```

==========

qual07.v1

```
Begin Execution at Fri Apr  4 11:53:01 2003


-- using default substitutions

select
supp_nation,
cust_nation,
l_year,
sum(volume) as revenue
from
(
select
n1.n_name as supp_nation,
n2.n_name as cust_nation,
to_number (to_char (l_shipdate,'yyyy')) as l_year,
l_extendedprice * (1 - l_discount) as volume
from
supplier,
lineitem,
orders,
customer,
nation n1,
nation n2
where
s_suppkey = l_suppkey
and o_orderkey = l_orderkey
and c_custkey = o_custkey
and s_nationkey = n1.n_nationkey
and c_nationkey = n2.n_nationkey
and (
(n1.n_name = 'FRANCE' and n2.n_name = 'GERMANY')
or (n1.n_name = 'GERMANY' and n2.n_name = 'FRANCE')
)
and l_shipdate between to_date( '1995-01-01', 'YYYY-
MM-DD') and to_date( '1996-12-31', 'YYYY-MM-DD')
) shipping
group by
supp_nation,
cust_nation,
l_year
order by
supp_nation,
cust_nation,
l_year

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


4 rows processed.
Statement Processed in 14.17 seconds.

Ended Executing this Stream at Fri Apr  4 11:53:15
2003


Stream Started at 1049485981.46
Stream Ended at 1049485995.63
Stream Processed in 14.17 seconds


SQL statements processed: 1
```

==========

qual08.v1

==========

```
Begin Execution at Fri Apr  4 11:53:15 2003


-- using default substitutions

select
o_year,
sum(case when nation='BRAZIL' then volume else 0 end
)/ sum(volume)
as mkt_share
from
(
select
to_number (to_char (o_orderdate, 'yyyy')) as o_year,
l_extendedprice * (1 - l_discount) as volume,
n2.n_name as nation
from
part,
supplier,
lineitem,
orders,
customer,
nation n1,
nation n2,
region
where
p_partkey = l_partkey
and s_suppkey = l_suppkey
and l_orderkey = o_orderkey
and o_custkey = c_custkey
and c_nationkey = n1.n_nationkey
and n1.n_regionkey = r_regionkey
and r_name = 'AMERICA'
and s_nationkey = n2.n_nationkey
and o_orderdate between to_date ('1995-01-01', 'YYYY-
MM-DD') and to_date ('1996-12-31', 'YYYY-MM-DD')
and p_type = 'ECONOMY ANODIZED STEEL'
) all_nations
group by
o_year
order by
```

```
o_year

O_YEAR                  MKT_SHARE
1995.00                 0.03
1996.00                 0.04


2 rows processed.
Statement Processed in 16.79 seconds.

Ended Executing this Stream at Fri Apr  4 11:53:32
2003


Stream Started at 1049485995.80
Stream Ended at 1049486012.59
Stream Processed in 16.79 seconds


SQL statements processed: 1
====================================================
```

qual09.v1

```
Begin Execution at Fri Apr  4 11:53:32 2003


-- using default substitutions

select
nation,
o_year,
sum(amount) as sum_profit
from
(
select
n_name as nation,
to_number (to_char (o_orderdate, 'yyyy')) as o_year,
l_extendedprice * (1 - l_discount) - ps_supplycost *
l_quantity as amount
from
part,
supplier,
lineitem,
partsupp,
orders,
nation
where
s_suppkey = l_suppkey
and ps_suppkey = l_suppkey
and ps_partkey = l_partkey
and p_partkey = l_partkey
and o_orderkey = l_orderkey
and s_nationkey = n_nationkey
and p_name like '%green%'
) profit
group by
nation,
o_year
order by
nation,
o_year desc

NATION                    O_YEAR
SUM_PROFIT
ALGERIA                   1998.00
31342867.23
ALGERIA                   1997.00
57138193.02
ALGERIA                   1996.00
56140140.13
ALGERIA                   1995.00
53051469.65
ALGERIA                   1994.00
53867582.13
ALGERIA                   1993.00
54942718.13
ALGERIA                   1992.00
54628034.71
ARGENTINA                 1998.00
30211185.71
ARGENTINA                 1997.00
50805741.75
ARGENTINA                 1996.00
51923746.58
ARGENTINA                 1995.00
49298625.77
ARGENTINA                 1994.00
50835610.11
ARGENTINA                 1993.00
51646079.18
ARGENTINA                 1992.00
50410314.99
... rows truncated...
UNITED STATES             1997.00
50077306.42
UNITED STATES             1996.00
48048649.47
UNITED STATES             1995.00
48809032.42
UNITED STATES             1994.00
49296747.18
UNITED STATES             1993.00
48029946.80
UNITED STATES             1992.00
48671944.50
VIETNAM                   1998.00
30442736.06
VIETNAM                   1997.00
50309179.79
VIETNAM                   1996.00
50488161.41
VIETNAM                   1995.00
49658284.61
VIETNAM                   1994.00
50596057.26
VIETNAM                   1993.00
50953919.15
VIETNAM                   1992.00
49613838.32


175 rows processed.
Statement Processed in 33.26 seconds.

Ended Executing this Stream at Fri Apr  4 11:54:06
2003


Stream Started at 1049486012.76
Stream Ended at 1049486046.01
Stream Processed in 33.26 seconds


SQL statements processed: 1
====================================================
```

qual10.v1

```
====================================================
Begin Execution at Fri Apr  4 11:54:06 2003


-- using default substitutions

select  *  from  (
select
c_custkey,
c_name,
sum(l_extendedprice * (1 - l_discount)) as revenue,
c_acctbal,
n_name,
c_address,
c_phone,
c_comment
from
customer,
orders,
lineitem,
nation
where
c_custkey = o_custkey
and l_orderkey = o_orderkey
and o_orderdate >= to_date ('1993-10-01',  'YYYY-MM-
DD')
and o_orderdate < add_months(  to_date( '1993-10-01',
'YYYY-MM-DD'),  3)
and l_returnflag = 'R'
and c_nationkey = n_nationkey
group by
c_custkey,
c_name,
c_acctbal,
c_phone,
```

```
n_name,
c_address,
c_comment
order by
revenue desc)
where rownum <= 20

C_CUSTKEY              C_NAME
REVENUE
C_ACCTBAL              N_NAME
C_ADDRESS                                C_PHONE
C_COMMENT
57040.00               Customer#000057040
734235.25
632.87                 JAPAN
nICtsILWBB
22-895-641-3466
requests sleep blithely about the furiously i
143347.00              Customer#000143347
721002.69
2557.47                EGYPT
,Q9Ml3w0gvX
14-742-935-3718
fluffily bold excuses haggle finally after the u
60838.00               Customer#000060838
679127.31
2454.77                BRAZIL
VWmQhWweqj5hFpcvhGFBeOY9hJ4m
12-913-494-9813
furiously even pinto beans integrate under the
ruthless foxes; ironic, even dolphins across the slyl
101998.00              Customer#000101998
637029.57
3790.89                UNITED KINGDOM
0,ORojfDdyMca2E2H
33-593-865-6378
accounts doze blithely! enticing, final deposits sleep
blithely special accounts. slyly express accounts pla
125341.00              Customer#000125341
633508.09
4983.51                GERMANY
9YRcnoUPOM7Sa8xymhsDHdQg
17-582-695-5962
quickly express requests wake quickly blithely
25501.00               Customer#000025501
620269.78
7725.04                ETHIOPIA
sr4VVVe3xCJQ2oo2QEhi19D,pXqo6kOGaSn2
15-874-808-6793
quickly special requests sleep evenly among the
special deposits. special deposi
115831.00              Customer#000115831
596423.87
5098.10                FRANCE
AlMpPnmtGrOFrDMUs5VLo EIA,Cg,Rw5TBuBoKiO
16-715-386-3788
carefully bold excuses sleep alongside of the thinly
idle
84223.00               Customer#000084223
594998.02
528.65                 UNITED KINGDOM
Eq51o UpQ4RBr  fYTdrZApDsPV4pQyuPq
33-442-824-8191
pending, final ideas haggle final requests. unusual,
regular asymptotes affix according to the even foxes.
54289.00               Customer#000054289
585603.39
5583.02                IRAN
x3ouCpz6,pRNVhajr0CCQG1
20-834-292-4707
express requests sublate blithely regular requests.
regular, even ideas solve.
39922.00               Customer#000039922
584878.11
7321.11                GERMANY
2KtWzW,FYkhdWBfobp6SFXWYKjvU9
17-147-757-8036
even pinto beans haggle. slyly bold accounts inte
6226.00                Customer#000006226
576783.76
2230.09                UNITED KINGDOM
TKbxS1dbkGMtaa,KOi26lbip4P0tPbWK0
33-657-701-3391
quickly final requests against the regular
instructions wake blithely final instructions. pa
922.00                 Customer#000000922
576767.53
3869.25                GERMANY
rsR9lRxyTdHbDOVt8nYbwjK5vAWH9sB
17-945-916-9648
boldly final requests cajole blith
147946.00              Customer#000147946
576455.13
2030.13                ALGERIA
Jqdt1kHAJtuTqHQK,B7 3tJh
10-886-956-3143
furiously even accounts are blithely above the
furiousl
115640.00              Customer#000115640
569341.19
6436.10                ARGENTINA
6yKLIRRAirUmBjKNO6Z3
11-411-543-4901
final instructions are slyly according to the
73606.00               Customer#000073606
568656.86
1785.67                JAPAN
vx9,7ACVtoKnLcoAHGNYDF
22-437-653-6966
furiously bold orbits about the furiously busy
requests wake across the furiously quiet theodolites.
d
110246.00              Customer#000110246
566842.98
7763.35                VIETNAM
UgsLFL3rendATzcHi
31-943-426-9837
dolphins sleep blithely among the slyly final
142549.00              Customer#000142549
563537.24
5085.99                INDONESIA
pJAmChWXct HNjPzgoBUOgAHduwwIR
19-955-562-2398
regular, unusual dependencies boost slyly; ironic
attainments nag fluffily into the unusual packages?
146149.00              Customer#000146149
557254.99
1791.55                ROMANIA
 STLwtlaB6
29-744-164-6487
silent, unusual requests detect quickly slyly regul
52528.00               Customer#000052528
556397.35
551.79                 ARGENTINA
elsyt8c9Z,7ch
11-208-192-3205
unusual requests detect. slyly dogged theodolites use
slyly. deposit
23431.00               Customer#000023431
554269.54
3381.86                ROMANIA
kKI5,CJAJQjQRQtOdCiFQ
29-915-458-2654
instructions nag quickly. furiously bold accounts
cajol


20 rows processed.
Statement Processed in 14.96 seconds.

Ended Executing this Stream at Fri Apr  4 11:54:21
2003


Stream Started at 1049486046.18
Stream Ended at 1049486061.14
Stream Processed in 14.96 seconds


SQL statements processed: 1
```

qual11.v1

```
Begin Execution at Fri Apr  4 11:54:21 2003


-- using default substitutions

select
ps_partkey,
sum(ps_supplycost * ps_availqty) as value
from
```

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

PS_PARTKEY	VALUE
129760.00	17538456.86
166726.00	16503353.92
191287.00	16474801.97
161758.00	16101755.54
34452.00	15983844.72
139035.00	15907078.34
9403.00	15451755.62
154358.00	15212937.88
38823.00	15064802.86
85606.00	15053957.15
33354.00	14408297.40
154747.00	14407580.68
82865.00	14235489.78
76094.00	14094247.04
222.00	13937777.74
121271.00	13908336.00
55221.00	13716120.47
22819.00	13666434.28
76281.00	13646853.68
85298.00	13581154.93
85158.00	13554904.00
139684.00	13535538.72
31034.00	13498025.25
87305.00	13482847.04
10181.00	13445148.75
62323.00	13411824.30
26489.00	13377256.38
96493.00	13339057.83
56548.00	13329014.97
55576.00	13306843.35
159751.00	13306614.48
92406.00	13287414.50
182636.00	13223726.74
199969.00	13135288.21
62865.00	13001926.94
7284.00	12945298.19
197867.00	12944510.52
11562.00	12931575.51
75165.00	12916918.12
97175.00	12911283.50
140840.00	12896562.23
65241.00	12890600.46
166120.00	12876927.22
9035.00	12863828.70
144616.00	12853549.30
176723.00	12832309.74
170884.00	12792136.58
29790.00	12723300.33
95213.00	12555483.73
183873.00	12550533.05
171235.00	12476538.30
21533.00	12437821.32
17290.00	12432159.50
156397.00	12260623.50
122611.00	12222812.98
139155.00	12220319.25
146316.00	12215800.61
171381.00	12199734.52
198633.00	12078226.95
167417.00	12046637.62
59512.00	12043468.76
31688.00	12034893.64
159586.00	12001505.84
8993.00	11963814.30
120302.00	11857707.55
43536.00	11779340.52
9552.00	11776909.16
86223.00	11772205.08
53776.00	11758669.65
131285.00	11616953.74
91628.00	11611114.83
169644.00	11567959.72
182299.00	11567462.05
33107.00	11453818.76
104184.00	11436657.44
67027.00	11419127.14
176869.00	11371451.71
30885.00	11369674.79
54420.00	11345076.88
72240.00	11313951.05
178708.00	11294635.17
81298.00	11273686.13
158324.00	11243442.72
117095.00	11242535.24
176793.00	11237733.38
86091.00	11177793.79
116033.00	11145434.36
129058.00	11119112.20
193714.00	11104706.39
117195.00	11077217.96
49851.00	11043701.78
19791.00	11030662.62
75800.00	11012401.62
161562.00	10996371.69
10119.00	10980015.75
39185.00	10970042.56
47223.00	10950022.13
175594.00	10942923.05
111295.00	10893675.61
155446.00	10852764.57
156391.00	10839810.38
.... rows truncated	
30802.00	8003710.88
107418.00	8000430.30
46620.00	7999778.35
191803.00	7994734.15
106343.00	7993087.76
59362.00	7990397.46
8329.00	7990052.90
75133.00	7988244.00
179023.00	7986829.62
135899.00	7985726.64
5824.00	7985340.02
148579.00	7984889.56
95888.00	7984735.72
9791.00	7982699.79
170437.00	7982370.72
39782.00	7977858.24
20605.00	7977556.00
28682.00	7976960.00
42172.00	7973399.00
56137.00	7971405.40
64729.00	7970769.72
98643.00	7968603.73
153787.00	7967535.58
8932.00	7967222.19
20134.00	7965713.28
197635.00	7963507.58
80408.00	7963312.17
37728.00	7961875.68
26624.00	7961772.31
44736.00	7961144.10
29763.00	7960605.03
36147.00	7959463.68
146040.00	7957587.66
115469.00	7957485.14
142276.00	7956790.63
181280.00	7954037.35
115096.00	7953047.55
109650.00	7952258.73
93862.00	7951992.24
158325.00	7950728.30
55952.00	7950387.06
122397.00	7947106.27
28114.00	7946945.72
11966.00	7945197.48
47814.00	7944083.00

```
85096.00                7943691.06
51657.00                7943593.77
196680.00               7943578.89
13141.00                7942730.34
193327.00               7941036.25
152612.00               7940663.71
139680.00               7939242.36
31134.00                7938318.30
45636.00                7937240.85
56694.00                7936015.95
8114.00                 7933921.88
71518.00                7932261.69
72922.00                7930400.64
146699.00               7929167.40
92387.00                7928972.67
186289.00               7928786.19
95952.00                7927972.78
196514.00               7927180.70
4403.00                 7925729.04
2267.00                 7925649.37
45924.00                7925047.68
11493.00                7916722.23
104478.00               7916253.60
166794.00               7913842.00
161995.00               7910874.27
23538.00                7909752.06
41093.00                7909579.92
112073.00               7908617.57
92814.00                7908262.50
88919.00                7907992.50
79753.00                7907933.88
108765.00               7905338.98
146530.00               7905336.60
71475.00                7903367.58
36289.00                7901946.50
61739.00                7900794.00
52338.00                7898638.08
194299.00               7898421.24
105235.00               7897829.94
77207.00                7897752.72
96712.00                7897575.27
10157.00                7897046.25
171154.00               7896814.50
79373.00                7896186.00
113808.00               7893353.88
27901.00                7892952.00
128820.00               7892882.72
25891.00                7890511.20
122819.00               7888881.02
154731.00               7888301.33
101674.00               7879324.60
51968.00                7879102.21
72073.00                7877736.11
5182.00                 7874521.73


1048 rows processed.
Statement Processed in 9.48 seconds.

Ended Executing this Stream at Fri Apr  4 11:54:30
2003


Stream Started at 1049486061.31
Stream Ended at 1049486070.79
Stream Processed in 9.48 seconds


SQL statements processed: 1
====================================================
```

qual12.v1

```
====================================================
Begin Execution at Fri Apr  4 11:54:30 2003


-- using default substitutions

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

L_SHIPMODE HIGH_LINE_COUNT        LOW_LINE_COUNT
MAIL       6202.00                9324.00
SHIP       6200.00                9262.00


2 rows processed.
Statement Processed in 8.87 seconds.

Ended Executing this Stream at Fri Apr  4 11:54:39
2003


Stream Started at 1049486070.96
Stream Ended at 1049486079.83
Stream Processed in 8.87 seconds


SQL statements processed: 1
====================================================
```

qual13.v1

```
====================================================
Begin Execution at Fri Apr  4 11:54:41 2003


-- using default substitutions

select
c_count,
count(*) as custdist
from
(
select
c_custkey,
count(o_orderkey)  as  c_count
from
customer, orders where
c_custkey = o_custkey(+)
and o_comment(+) not like '%special%requests%'
group by
c_custkey
) c_orders
group by
c_count
order by
custdist desc,
c_count desc

C_COUNT                CUSTDIST
0.00                   50004.00
9.00                   6641.00
10.00                  6566.00
11.00                  6058.00
8.00                   5949.00
12.00                  5553.00
13.00                  4989.00
19.00                  4748.00
7.00                   4707.00
18.00                  4625.00
15.00                  4552.00
17.00                  4530.00
14.00                  4484.00
20.00                  4461.00
```

```
16.00                   4323.00
21.00                   4217.00
22.00                   3730.00
6.00                    3334.00
23.00                   3129.00
24.00                   2622.00
25.00                   2079.00
5.00                    1972.00
26.00                   1593.00
27.00                   1185.00
4.00                    1033.00
28.00                   869.00
29.00                   559.00
3.00                    398.00
30.00                   373.00
31.00                   235.00
2.00                    144.00
32.00                   128.00
33.00                   71.00
34.00                   48.00
35.00                   33.00
1.00                    23.00
36.00                   17.00
37.00                   7.00
40.00                   4.00
38.00                   4.00
39.00                   2.00
41.00                   1.00


42 rows processed.
Statement Processed in 5.48 seconds.

Ended Executing this Stream at Fri Apr  4 11:54:46
2003


Stream Started at 1049486081.08
Stream Ended at 1049486086.56
Stream Processed in 5.48 seconds


SQL statements processed: 1
==================================================
```

qual14.v1

```
==================================================
```

Begin Execution at Fri Apr 4 11:54:46 2003

-- using default substitutions

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

PROMO_REVENUE
16.38


1 row# processed.
Statement Processed in 2.52 seconds.

Ended Executing this Stream at Fri Apr  4 11:54:49 2003


Stream Started at 1049486086.73
Stream Ended at 1049486089.25
Stream Processed in 2.52 seconds


SQL statements processed: 1
==================================================
```

qual15.v1

```
==================================================
```

Begin Execution at Fri Apr 4 11:54:49 2003

-- using default substitutions

```
with revenue as (
select
l_suppkey supplier_no,
sum(l_extendedprice * (1-l_discount)) total_revenue
from
lineitem
where
l_shipdate >= to_date ( '1996-01-01', 'YYYY-MM-DD')
and l_shipdate < add_months(to_date ( '1996-01-01', 'YYYY-MM-DD'), 3)
group by
l_suppkey
)
select
s_suppkey,
s_name,
s_address,
s_phone,
total_revenue
from
supplier,
revenue
where
s_suppkey = supplier_no
and total_revenue = (
select
max(total_revenue)
from
revenue
)
order by
s_suppkey

S_SUPPKEY           S_NAME
S_ADDRESS                       S_PHONE        TOTAL_REVENUE
8449.00             Supplier#000008449
5BXWsJERA2mP5OyO4                   20-469-856-8873 1772627.21


1 row# processed.
Statement Processed in 8.39 seconds.

Ended Executing this Stream at Fri Apr  4 11:54:57 2003


Stream Started at 1049486089.41
Stream Ended at 1049486097.80
Stream Processed in 8.39 seconds


SQL statements processed: 1
==================================================
```

qual16.v1

```
Begin Execution at Fri Apr  4 11:54:57 2003


-- using default substitutions

select
p_brand,
p_type,
p_size,
count(distinct ps_suppkey) as supplier_cnt
from
partsupp,
part
where
p_partkey = ps_partkey
and p_brand <> 'Brand#45'
and p_type not like 'MEDIUM POLISHED%'
```

```
and p_size in (49, 14, 23, 45, 19, 3, 36, 9)
and ps_suppkey not in (
select
s_suppkey
from
supplier
where
s_comment like '%Customer%Complaints%'
)
group by
p_brand,
p_type,
p_size
order by
supplier_cnt desc,
p_brand,
p_type,
p_size
```

P_BRAND	P_TYPE	P_SIZE	SUPPLIER_CNT
Brand#41	MEDIUM BRUSHED TIN	3.00	28.00
Brand#54	STANDARD BRUSHED COPPER	14.00	27.00
Brand#11	STANDARD BRUSHED TIN	23.00	24.00
Brand#11	STANDARD BURNISHED BRASS	36.00	24.00
Brand#15	MEDIUM ANODIZED NICKEL	3.00	24.00
Brand#15	SMALL ANODIZED BRASS	45.00	24.00
Brand#15	SMALL BURNISHED NICKEL	19.00	24.00
Brand#21	MEDIUM ANODIZED COPPER	3.00	24.00
Brand#22	SMALL BRUSHED NICKEL	3.00	24.00
Brand#22	SMALL BURNISHED BRASS	19.00	24.00
Brand#25	MEDIUM BURNISHED COPPER	36.00	24.00
Brand#31	PROMO POLISHED COPPER	36.00	24.00
Brand#33	LARGE POLISHED TIN	23.00	24.00
Brand#33	PROMO POLISHED STEEL	14.00	24.00
Brand#35	PROMO BRUSHED NICKEL	14.00	24.00
Brand#41	ECONOMY BRUSHED STEEL	9.00	24.00
Brand#41	ECONOMY POLISHED TIN	19.00	24.00
Brand#41	LARGE PLATED COPPER	36.00	24.00
Brand#42	ECONOMY PLATED BRASS	3.00	24.00
Brand#42	STANDARD POLISHED TIN	49.00	24.00
Brand#43	PROMO BRUSHED TIN	3.00	24.00
Brand#43	SMALL ANODIZED COPPER	36.00	24.00
Brand#44	STANDARD POLISHED NICKEL	3.00	24.00
Brand#52	ECONOMY PLATED TIN	14.00	24.00
Brand#52	STANDARD BURNISHED NICKEL	3.00	24.00
Brand#53	MEDIUM ANODIZED STEEL	14.00	24.00
Brand#14	PROMO ANODIZED NICKEL	45.00	23.00
Brand#32	ECONOMY PLATED BRASS	9.00	23.00
Brand#52	SMALL ANODIZED COPPER	3.00	23.00
Brand#11	ECONOMY BRUSHED COPPER	45.00	20.00
Brand#11	ECONOMY PLATED BRASS	23.00	20.00
Brand#11	LARGE BRUSHED COPPER	49.00	20.00
Brand#11	LARGE POLISHED COPPER	49.00	20.00
Brand#12	STANDARD ANODIZED TIN	49.00	20.00
Brand#12	STANDARD PLATED BRASS	19.00	20.00
Brand#13	ECONOMY BRUSHED BRASS	9.00	20.00
Brand#13	ECONOMY BURNISHED STEEL	14.00	20.00
Brand#13	LARGE BURNISHED NICKEL	19.00	20.00
Brand#13	MEDIUM BURNISHED COPPER	36.00	20.00
Brand#13	SMALL BRUSHED TIN	45.00	20.00
Brand#13	STANDARD ANODIZED COPPER	3.00	20.00
Brand#13	STANDARD PLATED NICKEL	23.00	20.00
Brand#14	ECONOMY ANODIZED COPPER	14.00	20.00
Brand#14	ECONOMY PLATED TIN	36.00	20.00
Brand#14	ECONOMY POLISHED NICKEL	3.00	20.00
Brand#14	MEDIUM ANODIZED NICKEL	3.00	20.00
Brand#14	SMALL POLISHED TIN	14.00	20.00
Brand#15	MEDIUM ANODIZED COPPER	9.00	20.00
Brand#15	MEDIUM PLATED TIN	23.00	20.00
Brand#15	PROMO PLATED BRASS	14.00	20.00
Brand#15	SMALL ANODIZED COPPER	45.00	20.00
Brand#15	SMALL PLATED COPPER	49.00	20.00
Brand#15	STANDARD PLATED TIN	3.00	20.00
Brand#21	LARGE ANODIZED COPPER	36.00	20.00
Brand#21	LARGE BRUSHED TIN	3.00	20.00
Brand#21	MEDIUM ANODIZED COPPER	14.00	20.00
Brand#21	PROMO BRUSHED TIN	36.00	20.00
Brand#21	PROMO POLISHED NICKEL	45.00	20.00
Brand#21	SMALL ANODIZED COPPER	9.00	20.00
Brand#21	SMALL POLISHED NICKEL	23.00	20.00
Brand#22	LARGE ANODIZED COPPER	36.00	20.00
Brand#22	LARGE BRUSHED COPPER	49.00	20.00
Brand#22	PROMO ANODIZED TIN	49.00	20.00
Brand#22	PROMO POLISHED BRASS	45.00	20.00
Brand#22	SMALL BURNISHED STEEL	45.00	20.00
Brand#23	MEDIUM ANODIZED STEEL	45.00	20.00
Brand#23	PROMO POLISHED STEEL	23.00	20.00
Brand#23	STANDARD BRUSHED TIN	14.00	20.00
Brand#23	STANDARD PLATED NICKEL	36.00	20.00
Brand#24	PROMO PLATED COPPER	49.00	20.00
Brand#24	PROMO PLATED STEEL	49.00	20.00
Brand#24	PROMO POLISHED STEEL	9.00	20.00
Brand#24	STANDARD BRUSHED TIN	36.00	20.00
Brand#25	LARGE ANODIZED BRASS	3.00	20.00
Brand#25	PROMO BURNISHED TIN	3.00	20.00
Brand#31	ECONOMY POLISHED NICKEL	3.00	

```
20.00
Brand#31   MEDIUM PLATED TIN         45.00
20.00
Brand#31   SMALL ANODIZED STEEL      14.00
20.00
Brand#32   ECONOMY ANODIZED COPPER   36.00
20.00
Brand#32   ECONOMY BRUSHED NICKEL    49.00
20.00
Brand#32   LARGE ANODIZED TIN        19.00
20.00
Brand#32   MEDIUM BURNISHED COPPER   19.00
20.00
Brand#32   SMALL ANODIZED STEEL      45.00
20.00
Brand#33   ECONOMY POLISHED COPPER   19.00
20.00
Brand#33   PROMO PLATED NICKEL       14.00
20.00
Brand#33   SMALL POLISHED TIN        9.00
20.00
Brand#33   STANDARD ANODIZED BRASS   49.00
20.00
Brand#33   STANDARD BURNISHED BRASS  45.00
20.00
Brand#34   ECONOMY BRUSHED NICKEL    49.00
20.00
Brand#34   LARGE BRUSHED BRASS       19.00
20.00
Brand#34   SMALL BRUSHED TIN         3.00
20.00
Brand#34   STANDARD PLATED COPPER    9.00
20.00
Brand#35   LARGE ANODIZED NICKEL     3.00
20.00
Brand#35   MEDIUM ANODIZED BRASS     45.00
20.00
Brand#35   MEDIUM ANODIZED STEEL     23.00
20.00
Brand#35   PROMO ANODIZED COPPER     49.00
20.00
Brand#35   SMALL POLISHED COPPER     14.00
20.00
Brand#41   LARGE ANODIZED STEEL      3.00
20.00
Brand#41   LARGE BRUSHED NICKEL      23.00
20.00
Brand#41   LARGE BURNISHED COPPER    3.00
20.00
Brand#41   MEDIUM PLATED STEEL       19.00
20.00
..... rows truncated ....
Brand#55   SMALL POLISHED COPPER     36.00
4.00
Brand#55   SMALL POLISHED COPPER     45.00
4.00
Brand#55   SMALL POLISHED COPPER     49.00
4.00
Brand#55   SMALL POLISHED NICKEL     9.00
4.00
Brand#55   SMALL POLISHED NICKEL     14.00
4.00
Brand#55   SMALL POLISHED NICKEL     19.00
4.00
Brand#55   SMALL POLISHED NICKEL     23.00
4.00
Brand#55   SMALL POLISHED NICKEL     45.00
4.00
Brand#55   SMALL POLISHED NICKEL     49.00
4.00
Brand#55   SMALL POLISHED STEEL      19.00
4.00
Brand#55   SMALL POLISHED STEEL      45.00
4.00
Brand#55   SMALL POLISHED TIN        14.00
4.00
Brand#55   SMALL POLISHED TIN        23.00
4.00
Brand#55   SMALL POLISHED TIN        45.00
4.00
Brand#55   STANDARD ANODIZED BRASS   9.00
4.00
Brand#55   STANDARD ANODIZED BRASS   23.00
4.00
Brand#55   STANDARD ANODIZED BRASS   49.00
4.00
Brand#55   STANDARD ANODIZED COPPER  9.00
4.00
Brand#55   STANDARD ANODIZED COPPER  14.00
4.00
Brand#55   STANDARD ANODIZED COPPER  45.00
4.00
Brand#55   STANDARD ANODIZED NICKEL  3.00
4.00
Brand#55   STANDARD ANODIZED NICKEL  14.00
4.00
Brand#55   STANDARD ANODIZED NICKEL  45.00
4.00
Brand#55   STANDARD ANODIZED NICKEL  49.00
4.00
Brand#55   STANDARD ANODIZED STEEL   3.00
4.00
Brand#55   STANDARD ANODIZED STEEL   14.00
4.00
Brand#55   STANDARD ANODIZED TIN     14.00
4.00
Brand#55   STANDARD ANODIZED TIN     36.00
4.00
Brand#55   STANDARD ANODIZED TIN     45.00
4.00
Brand#55   STANDARD BRUSHED BRASS    9.00
4.00
Brand#55   STANDARD BRUSHED BRASS    19.00
4.00
Brand#55   STANDARD BRUSHED COPPER   14.00
4.00
Brand#55   STANDARD BRUSHED COPPER   19.00
4.00
Brand#55   STANDARD BRUSHED NICKEL   3.00
4.00
Brand#55   STANDARD BRUSHED NICKEL   36.00
4.00
Brand#55   STANDARD BRUSHED STEEL    9.00
4.00
Brand#55   STANDARD BRUSHED STEEL    14.00
4.00
Brand#55   STANDARD BRUSHED STEEL    19.00
4.00
Brand#55   STANDARD BRUSHED STEEL    49.00
4.00
Brand#55   STANDARD BRUSHED TIN      19.00
4.00
Brand#55   STANDARD BRUSHED TIN      49.00
4.00
Brand#55   STANDARD BURNISHED BRASS  9.00
4.00
Brand#55   STANDARD BURNISHED BRASS  19.00
4.00
Brand#55   STANDARD BURNISHED BRASS  23.00
4.00
Brand#55   STANDARD BURNISHED BRASS  36.00
4.00
Brand#55   STANDARD BURNISHED COPPER 3.00
4.00
Brand#55   STANDARD BURNISHED NICKEL 9.00
4.00
Brand#55   STANDARD BURNISHED NICKEL 49.00
4.00
Brand#55   STANDARD BURNISHED STEEL  19.00
4.00
Brand#55   STANDARD BURNISHED STEEL  23.00
4.00
Brand#55   STANDARD BURNISHED STEEL  36.00
4.00
Brand#55   STANDARD BURNISHED STEEL  45.00
4.00
Brand#55   STANDARD BURNISHED TIN    9.00
4.00
Brand#55   STANDARD BURNISHED TIN    19.00
4.00
Brand#55   STANDARD BURNISHED TIN    36.00
4.00
Brand#55   STANDARD BURNISHED TIN    49.00
4.00
Brand#55   STANDARD PLATED BRASS     9.00
4.00
Brand#55   STANDARD PLATED BRASS     45.00
4.00
Brand#55   STANDARD PLATED BRASS     49.00
4.00
Brand#55   STANDARD PLATED COPPER    9.00
4.00
```

```
Brand#55   STANDARD PLATED COPPER    45.00
4.00
Brand#55   STANDARD PLATED NICKEL    3.00
4.00
Brand#55   STANDARD PLATED NICKEL    19.00
4.00
Brand#55   STANDARD PLATED NICKEL    45.00
4.00
Brand#55   STANDARD PLATED STEEL     14.00
4.00
Brand#55   STANDARD PLATED STEEL     23.00
4.00
Brand#55   STANDARD PLATED STEEL     49.00
4.00
Brand#55   STANDARD PLATED TIN       9.00
4.00
Brand#55   STANDARD PLATED TIN       14.00
4.00
Brand#55   STANDARD PLATED TIN       36.00
4.00
Brand#55   STANDARD POLISHED BRASS   3.00
4.00
Brand#55   STANDARD POLISHED BRASS   9.00
4.00
Brand#55   STANDARD POLISHED BRASS   23.00
4.00
Brand#55   STANDARD POLISHED COPPER  3.00
4.00
Brand#55   STANDARD POLISHED COPPER  23.00
4.00
Brand#55   STANDARD POLISHED COPPER  45.00
4.00
Brand#55   STANDARD POLISHED NICKEL  3.00
4.00
Brand#55   STANDARD POLISHED NICKEL  23.00
4.00
Brand#55   STANDARD POLISHED NICKEL  36.00
4.00
Brand#55   STANDARD POLISHED NICKEL  45.00
4.00
Brand#55   STANDARD POLISHED NICKEL  49.00
4.00
Brand#55   STANDARD POLISHED STEEL   14.00
4.00
Brand#55   STANDARD POLISHED STEEL   23.00
4.00
Brand#55   STANDARD POLISHED TIN     9.00
4.00
Brand#55   STANDARD POLISHED TIN     19.00
4.00
Brand#55   STANDARD POLISHED TIN     36.00
4.00
Brand#11   SMALL BRUSHED TIN         19.00
3.00
Brand#15   LARGE PLATED NICKEL       45.00
3.00
Brand#15   LARGE POLISHED NICKEL     9.00
3.00
Brand#21   PROMO BURNISHED STEEL     45.00
3.00
Brand#22   STANDARD PLATED STEEL     23.00
3.00
Brand#25   LARGE PLATED STEEL        19.00
3.00
Brand#32   STANDARD ANODIZED COPPER  23.00
3.00
Brand#33   SMALL ANODIZED BRASS      9.00
3.00
Brand#35   MEDIUM ANODIZED TIN       19.00
3.00
Brand#51   SMALL PLATED BRASS        23.00
3.00
Brand#52   MEDIUM BRUSHED BRASS      45.00
3.00
Brand#53   MEDIUM BRUSHED TIN        45.00
3.00
Brand#54   ECONOMY POLISHED BRASS    9.00
3.00
Brand#55   PROMO PLATED BRASS        19.00
3.00
Brand#55   STANDARD PLATED TIN       49.00
3.00


18314 rows processed.
Statement Processed in 5.73 seconds.
```

```
Ended Executing this Stream at Fri Apr  4 11:55:03
2003


Stream Started at 1049486097.96
Stream Ended at 1049486103.69
Stream Processed in 5.73 seconds


SQL statements processed: 1

==================================================
```

qual17.v1

```
==================================================
Begin Execution at Fri Apr  4 11:55:03 2003


-- using default substitutions

select
sum(l_extendedprice) / 7.0 as avg_yearly
from
lineitem,
part
where
p_partkey = l_partkey
and p_brand = 'Brand#23'
and p_container = 'MED BOX'
and l_quantity < (
select
0.2 * avg(l_quantity)
from
lineitem
where
l_partkey = p_partkey
)

AVG_YEARLY
348406.05


1 row# processed.
Statement Processed in 14.00 seconds.

Ended Executing this Stream at Fri Apr  4 11:55:17 2003


Stream Started at 1049486103.86
Stream Ended at 1049486117.85
Stream Processed in 14.00 seconds


SQL statements processed: 1
==================================================
```

qual18.v1

```
==================================================
Begin Execution at Fri Apr  4 11:55:18 2003


-- using default substitutions

select  *  from  (
select
c_name,
c_custkey,
o_orderkey,
o_orderdate,
o_totalprice,
sum(l_quantity)
from
customer,
orders,
lineitem
where
o_orderkey in (
select
l_orderkey
from
lineitem
```

```
group by
l_orderkey having
sum(l_quantity) > 300
)
and c_custkey = o_custkey
and o_orderkey = l_orderkey
group by
c_name,
c_custkey,
o_orderkey,
o_orderdate,
o_totalprice
order by
o_totalprice desc,
o_orderdate
)
where rownum <= 100
```

C_NAME	C_CUSTKEY
O_ORDERKEY	O_ORDERDATE
O_TOTALPRICE	SUM(L_QUANTITY)
Customer#000128120	128120.00
4722021.00	1994-04-07
544089.09	323.00
Customer#000144617	144617.00
3043270.00	1997-02-12
530604.44	317.00
Customer#000013940	13940.00
2232932.00	1997-04-13
522720.61	304.00
Customer#000066790	66790.00
2199712.00	1996-09-30
515531.82	327.00
Customer#000046435	46435.00
4745607.00	1997-07-03
508047.99	309.00
Customer#000015272	15272.00
3883783.00	1993-07-28
500241.33	302.00
Customer#000146608	146608.00
3342468.00	1994-06-12
499794.58	303.00
Customer#000096103	96103.00
5984582.00	1992-03-16
494398.79	312.00
Customer#000024341	24341.00
1474818.00	1992-11-15
491348.26	302.00
Customer#000137446	137446.00
5489475.00	1997-05-23
487763.25	311.00
Customer#000107590	107590.00
4267751.00	1994-11-04
485141.38	301.00
Customer#000050008	50008.00
2366755.00	1996-12-09
483891.26	302.00
Customer#000015619	15619.00
3767271.00	1996-08-07
480083.96	318.00
Customer#000077260	77260.00
1436544.00	1992-09-12
479499.43	307.00
Customer#000109379	109379.00
5746311.00	1996-10-10
478064.11	302.00
Customer#000054602	54602.00
5832321.00	1997-02-09
471220.08	307.00
Customer#000105995	105995.00
2096705.00	1994-07-03
469692.58	307.00
Customer#000148885	148885.00
2942469.00	1992-05-31
469630.44	313.00
Customer#000114586	114586.00
551136.00	1993-05-19
469605.59	308.00
Customer#000105260	105260.00
5296167.00	1996-09-06
469360.57	303.00
Customer#000147197	147197.00
1263015.00	1997-02-02
467149.67	320.00
Customer#000064483	64483.00
2745894.00	1996-07-04
466991.35	304.00
Customer#000136573	136573.00
2761378.00	1996-05-31
461282.73	301.00
Customer#000016384	16384.00
502886.00	1994-04-12
458378.92	312.00
Customer#000117919	117919.00
2869152.00	1996-06-20
456815.92	317.00
Customer#000012251	12251.00
735366.00	1993-11-24
455107.26	309.00
Customer#000120098	120098.00
1971680.00	1995-06-14
453451.23	308.00
Customer#000066098	66098.00
5007490.00	1992-08-07
453436.16	304.00
Customer#000117076	117076.00
4290656.00	1997-02-05
449545.85	301.00
Customer#000129379	129379.00
4720454.00	1997-06-07
448665.79	303.00
Customer#000126865	126865.00
4702759.00	1994-11-07
447606.65	320.00
Customer#000088876	88876.00
983201.00	1993-12-30
446717.46	304.00
Customer#000036619	36619.00
4806726.00	1995-01-17
446704.09	328.00
Customer#000141823	141823.00
2806245.00	1996-12-29
446269.12	310.00
Customer#000053029	53029.00
2662214.00	1993-08-13
446144.49	302.00
Customer#000018188	18188.00
3037414.00	1995-01-25
443807.22	308.00
Customer#000066533	66533.00
29158.00	1995-10-21
443576.50	305.00
Customer#000037729	37729.00
4134341.00	1995-06-29
441082.97	309.00
Customer#000003566	3566.00
2329187.00	1998-01-04
439803.36	304.00
Customer#000045538	45538.00
4527553.00	1994-05-22
436275.31	305.00
Customer#000081581	81581.00
4739650.00	1995-11-04
435405.90	305.00
Customer#000119989	119989.00
1544643.00	1997-09-20
434568.25	320.00
Customer#000003680	3680.00
3861123.00	1998-07-03
433525.97	301.00
Customer#000113131	113131.00
967334.00	1995-12-15
432957.75	301.00
Customer#000141098	141098.00
565574.00	1995-09-24
430986.69	301.00
Customer#000093392	93392.00
5200102.00	1997-01-22
425487.51	304.00
Customer#000015631	15631.00
1845057.00	1994-05-12
419879.59	302.00
Customer#000112987	112987.00
4439686.00	1996-09-17
418161.49	305.00
Customer#000012599	12599.00
4259524.00	1998-02-12
415200.61	304.00
Customer#000105410	105410.00
4478371.00	1996-03-05
412754.51	302.00
Customer#000149842	149842.00

```
5156581.00              1994-05-30
411329.35               302.00
Customer#000010129         10129.00
5849444.00              1994-03-21
409129.85               309.00
Customer#000069904         69904.00
1742403.00              1996-10-19
408513.00               305.00
Customer#000017746         17746.00
6882.00                 1997-04-09
408446.93               303.00
Customer#000013072         13072.00
1481925.00              1998-03-15
399195.47               301.00
Customer#000082441         82441.00
857959.00               1994-02-07
382579.74               305.00
Customer#000088703         88703.00
2995076.00              1994-01-30
363812.12               302.00


57 rows processed.
Statement Processed in 13.79 seconds.

Ended Executing this Stream at Fri Apr  4 11:55:31
2003


Stream Started at 1049486118.01
Stream Ended at 1049486131.81
Stream Processed in 13.79 seconds


SQL statements processed: 1
```

==========

qual19.v1

==========

Begin Execution at Fri Apr 4 11:55:31 2003

-- using default substitutions

```
select
sum(l_extendedprice* (1 - l_discount)) as revenue
from
lineitem,
part
where
(
p_partkey = l_partkey
and p_brand = 'Brand#12'
and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
and l_quantity >= 1 and l_quantity <= 1 + 10
and p_size between 1 and 5
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
)
or
(
p_partkey = l_partkey
and p_brand = 'Brand#23'
and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
and l_quantity >= 10 and l_quantity <= 10 + 10
and p_size between 1 and 10
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
)
or
(
p_partkey = l_partkey
and p_brand = 'Brand#34'
and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
and l_quantity >= 20 and l_quantity <= 20 + 10
and p_size between 1 and 15
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
)

REVENUE
3083843.06


1 row# processed.
Statement Processed in 12.94 seconds.

Ended Executing this Stream at Fri Apr  4 11:55:44 2003


Stream Started at 1049486131.97
Stream Ended at 1049486144.92
Stream Processed in 12.94 seconds


SQL statements processed: 1
```

==========

qual20.v1

```
Begin Execution at Fri Apr  4 11:55:45 2003


-- using default substitutions

select
s_name,
s_address
from
supplier,
nation
where
s_suppkey in (
select
ps_suppkey
from
partsupp
where
ps_partkey in (
select
p_partkey
from
part
where
p_name like 'forest%'
)
and ps_availqty > (
select
0.5 * sum(l_quantity)
from
lineitem
where
l_partkey = ps_partkey
and l_suppkey = ps_suppkey
and l_shipdate >= to_date ('1994-01-01', 'YYYY-MM-DD')
and l_shipdate < add_months( to_date ('1994-01-01',
'YYYY-MM-DD'), 12)
)
)
and s_nationkey = n_nationkey
and n_name = 'CANADA'
order by
s_name

S_NAME                    S_ADDRESS
Supplier#000000020        JtPqm19E7tF 152Rl1wQZ8j0H
Supplier#000000091
35WVnU7GLNbQDcc2TARavGtk6RB6ZCd46UAY
Supplier#000000197        3oYqODDUGH3XsHXmPuzYHW5NLU3,
ONZl
Supplier#000000226        TXAcN7Ym29ObaqaKmyDrl91mN
Supplier#000000285        q TMZEDyZtv
vUiFKBhT3NJlnIxpL
Supplier#000000378        mLPJtpu4wOc cSFzBR
Supplier#000000402        JR8vWoCteJtJg3okRpt0r28KEo
Supplier#000000530
0BvoewCPg2scOEfuL93FRKqSxHmdhw1
Supplier#000000688
orLvVDOBBE2B2ppjbiTZTJoIJgn1VgKg3
Supplier#000000710
MZR2bxctPrbC2vHho0CbuOBkLJMpqOwghC
Supplier#000000736        GUIYDfv5xCxLgDx6KQ8khY
ntVVnFqmfMKIgT
Supplier#000000761        tF8fMGa6HY4
w77mDwT4rO21kxwe7uTSYNW
Supplier#000000884        QEJm8KMHQ8
```

Supplier#000000887 y mQ7NHjVbdqnbYr9 L
Supplier#000000935 JHRSOterYgt4MTNo7cupTzA, 6MoNw 4
Supplier#000000975 1qorM1ypBdwgPVuf6sMCKuF9D1rJN1iCTXKmalSt
Supplier#000001263 Aa4 UELS1JqY7qIjniwCWiC
Supplier#000001399 gE COEie8t3c4xqeuT
Supplier#000001446 GOJRjfd6Z9UQ,fiuPz6CO5GDGV
Supplier#000001454 8cBIELxY754iHJIgRXJ0
Supplier#000001500 wnElVvfuyaPJROy6x0
Supplier#000001602 ygd4iNQLQeVWW
Supplier#000001626 7Jud9t6xZNzlEB,
Supplier#000001682 C37Gkv 7a5ujZ9
Supplier#000001699 bRoW MhYUCJH5bABOA53N9i
Supplier#000001700 UJeFoDLZ2VtflK
Supplier#000001726 8M92T8y7jYXzmvCANTtqR8GHuT
Supplier#000001730 aOSM0,btPDs UsC06himJn, 6nswJG
Supplier#000001746 B4zMDOFcMGbkZQ4XxA, UyaQoEWZzMSGI wTsJp9N
Supplier#000001752 m QTCyyuWZ8j2hQa7JNclIpGr
Supplier#000001856 I3O9Kt2l,QK, VI9TBUvV9PhhfP6kt2J69bwmv64e
Supplier#000001931 mBhPe7YJU1aYMpwTiRdivl
Supplier#000001939 e G0hrWPL9N6z
Supplier#000001990 o8nhGpKphsybKZtye0o6OfDz9GIcuGHjMs7pq
Supplier#000002020 IpU ZOTeubVOC
Supplier#000002022 sNvMPl3TiNZBOYV3w2XX
Supplier#000002036 Z,ty7z5cPHh66iY5op,q
Supplier#000002204 y2EF XUo,UyNoAQEH, gIazC7aRG1zmuzzf
Supplier#000002243 E8cm5YhMc6UR
Supplier#000002245 KsYA4445HcugJAb3eCmvtUslGA7Qne
Supplier#000002282 n8YZgSNuC4, iZ7s5oHTMHNFdv94DwZ2rrUEb0pgD
Supplier#000002303 EoC4LCpU2cuEPKcKyTFyMGFBGkF
Supplier#000002373 asj8ud7aEmGoHuiqI5qVZ1rhpWS hJc9tF9
Supplier#000002419 BtNpaOZWiVGVE53RWL22
Supplier#000002481 Efgj6dDYeFS7cq, V1DARlmeOiGecY
Supplier#000002571 i16xKt,WOrJhlf GkzsRdrd04sZ 5jei9MtB
Supplier#000002585 pzbCgCvYax82Wq5, dG4xzyDMiRW8d
Supplier#000002630 2iaqwId62RhjVIwNrsETHr5FbJnFTopCYZ
Supplier#000002719 XDDsajYoPAama6xjz37p66GQbXEd8X
Supplier#000002721 k5NlqeYhjeb8BgE
Supplier#000002730 Gilu9XLsEX,oU0EyshvFTWs
Supplier#000002775 unOFQoQpnWJJj
Supplier#000002853 A7dqcjy3GQ
Supplier#000002875 VhKeIsRayU6EvfXgyyZ8aFgvv0HMfa19q1Q
Supplier#000002934 F0y pndtv8r vKXoJp
Supplier#000002941 eQNNPRrS27ngMG1ub,
Supplier#000002960 hobom 9kkCVwuS7uKjJU0Eq2LS 8ya8Y7
Supplier#000002980 qRGRt6ztQ3x4Gyc9jMsAkIqmBSIN,8hsGv
Supplier#000003062 g8adKB2ZuDdssRsqoQBtUjvc1b
Supplier#000003087 rdvMSbz1W LHZj8B5sRR,M4a
Supplier#000003089 VPr5pi1Ae915QrArKTf8NOKYSkP7mU iZu
Supplier#000003095 kucdXIhJ6IYsHy0ArE7n
Supplier#000003201 nSTtv Ui0y0BdzWb4T6snugIhEhn14yM
Supplier#000003213 Cu 9bXI aZ6CtLa1N7LX
Supplier#000003241 I,U861f8RcYEvHqfc5IqeMiJP
Supplier#000003275 SucWDuhYahP3UwkM
Supplier#000003288 nnNLdzTmV6P0uf7yBCiK3fWt2UxHJ0
Supplier#000003313 nFYiTwM0,WS8b wDE6EWK1vhCdJ1Gwgufh3L5j
Supplier#000003314 IDIz6TesAcXI6pXtzbO uzevXon6CH9WATfo
Supplier#000003380 Ibw, 51BzsCyxl7X3zqAFc0Gu7Yx0K7mImP9z0u
Supplier#000003403 NX3YCs0gokSyzjHJSqr34ouIlZJ14pw
Supplier#000003421 9JXNyS4VCMDLl4CxDlJ L0
Supplier#000003441 twlhit80C6y8JjHCO3
Supplier#000003590 tTRoffuAP1oPC
Supplier#000003607 GdAljb2Hv8rGL
Supplier#000003625 B2VSS5,2GVJQ6tZa37KdAmg erHp2jPgz
Supplier#000003656 Nm2EEdYKEnshNLkYY3ynLh50NQVU
Supplier#000003782 J5zb1LTPG,V7AdewI,fgQf6 o,sE K
Supplier#000003918 FM9w9pzhCqQLAN,aMW
Supplier#000003941 cEP,VFaLpe9UZScU4gA1irRwtx
Supplier#000003994 5,,f1CXdHg0
Supplier#000004005 6TYXl,votYNqWcKjxS7Hqyc56kB
Supplier#000004033 EOzqJxRhM0Hl37dILPY
Supplier#000004140 0KfTnit2HOkf
Supplier#000004165 w7hYN2daqScYCIRSdUn7XTDNju03gnZ
Supplier#000004207 zlUWBwIceWjG4HdYE80M0V4xqIfu
Supplier#000004236 OF0jbzhEkICu2z8BDRvEBGx4H y0EoNhScU8
Supplier#000004246 4JQrQ4alTxXA6Xf jn
Supplier#000004278 QpNNPCpui5CrnIR
Supplier#000004343 lgXzPDBA anG4fex5plol
Supplier#000004346 9X,TUgmcvC
Supplier#000004388 6L1rF,h0EvCWQ8
Supplier#000004406 rJ, 1QgxU5vyLb46t0TGP4K21JRxmQjBkiK
Supplier#000004430 uw8zge87fVkf3pm8WFxc bdfuhsqeG
Supplier#000004522 v3yogwz1nqAuipkoLs3YcKCPk2zpK
Supplier#000004527 CrB53pDuKOHF4lUrZCXck2XA4V
Supplier#000004542 eh8Pfin C2lYtZ Y nIgL
Supplier#000004574 ,jwkvE5xM1WoiDNP
Supplier#000004655 VTdApPWryXckXlSRcrizB4dAWHkQbCw2f
Supplier#000004701 VH3WxXT69sjejL
Supplier#000004711 QmsHB,aNaxsFzYmaeuw, HErwDVPxYs4FfZ
Supplier#000004987 s6luZyBhSexcwKlIqS
Supplier#000005000 oM3S,Xrv,jTl o6xQJKtsIFPxtp3Xdh
Supplier#000005100 dLw2b XiC0wmwwfkdQfxp3
Supplier#000005192 inBW K3Inv0HLU9k
Supplier#000005195 5CIXPZ1QIOcJr18MxZmL4Jm
Supplier#000005283 WLu23vv3t08a30eBnoYJvsta
Supplier#000005300 L3kYS3ABu6
Supplier#000005386 7PUqqMjB4f5b
Supplier#000005426 SnsYc67ZAr PWpgUZFHa, 3Hbp39bwc
Supplier#000005484 QlNcarA4Trl20XRzbAIJqQZZfCYYBpCd,pzb gC
Supplier#000005505 OYzPHlWXgvfb3
Supplier#000005506 dDrLVuBsC4Lp
Supplier#000005516 4zdRRhzRvmxOCJ6VH9nYfMMPamLTUEoSwCwy2
Supplier#000005536 esCRylGBKPk70m1Xn0TTf2gFWRJZo
Supplier#000005605 U5HUmJF,E7KAGe
Supplier#000005631 ,W75 IFsCY9hmp2pnKBew7Fwv9Ao
Supplier#000005730 W GP, c8MvzsjwuGgShQnXZ6BD9lYOKTjZ
Supplier#000005736 ZNAr382Jy258LB
Supplier#000005737 OEm4O9XYoXHu0N0qRVrcF2DWS
Supplier#000005797 1C0cMQv9P8oE5FRLdRLB5boIAqr6CKxF89
Supplier#000005836 zuX8HcnY1x5klp9k0
Supplier#000005875 Gg0z2JkspRXJ8tjuRuw82lP5aeo1MYg31xkQ8
Supplier#000005974 amJ9VIm0Ffyza3wMVW8v3t8Kz985lJy
Supplier#000005989 AHl20WGKfBqxTO
Supplier#000006059 XE,OwTevhRu3YwFwir1
Supplier#000006065 7IiYouX4W7yVzfGsfxw3g9tUgJJFkw
Supplier#000006070 8QFoVF,Bn dUn1PEKLEkFAM
Supplier#000006109 A2VKPMJXNgkDtFOb67bkpvDPM
Supplier#000006121 9RYtO5vms23vWkzCophd,4exjKC1
Supplier#000006215 IYImP8zF0Vc4NA42TGZtJi8PTByII1GHnibnC
Supplier#000006217 a5dYX927RHND6MQ5k36N
Supplier#000006274 9Xt715Au7g7ArKrbaKO4SrqAJpGd1z5VZJjyv6
Supplier#000006435 viKmUS3zs2QDcWmDDTOjkcSt
Supplier#000006463 UrvGNIYmcWSICyFNtYGEjerqnCf8zsl5X9d5H
Supplier#000006493 DH5rL1zdQpVjET 1LGDn67fUXAQhKqI1g
Supplier#000006521 QRrYsIjsu9

```
Supplier#000006607        YlrYMZKAnVyVp
Supplier#000006706        rGWKQ1MbxZb1UoXBG AHDzQ3,
LuwA8SwGPM
Supplier#000006761        EWHJuleApVC
nZjKBfwvA48ycgFFQ
Supplier#000006808        kkNY3DrRDmPjhJ1x3H3u5giBqC7
Supplier#000006858
MDFid8SSVwqpJz4w7kI10DYYyKvk2ZVJrkjiHYZ
Supplier#000006872        4inbIrTbf3p4gU8NmOFN
Supplier#000006949        Ffu26iJzkOgygMr1klI exZSXrw7
Supplier#000006985        c
66IQCaBt1cyKh0,1Wpvha6tAvROXj
Supplier#000007072
Zy9t3SeZQrX9OEVUzTTRmZqdkSHFBg
Supplier#000007098
lXHSK0hoWcPPqxYd5CbjA3a4ep6NHATvKojdmux
Supplier#000007135        GzrnCh5T5VyFLatS5
Supplier#000007160
8Ankp7fpXO8Ai7UmgnwESp8WMXw0sv3IabP
Supplier#000007169        zmORVoYECdS8SWDOVVc0OFD4
Supplier#000007322        w
TNKzMVArfqHI20yvEIYetnG9e3Z
Supplier#000007365        WZuJ9dfwaei,VnDOy14y
Supplier#000007398        6SMmUD1,,cmd60
Supplier#000007402        X65wVTM tZAHEA8aV
Supplier#000007448        uJJB4JhITmiUaV5pQa
Supplier#000007477        SERH,wLJ4spw5juH60bBruv8j0K
Supplier#000007509        BS05Ugh9CjiHjOcy8kTQg7eK
Supplier#000007561        AeOlKZVX,5p
Supplier#000007789        AaTO6Pb yxOctcX wd8GJEAjUB
L47Z8s
Supplier#000007801        VRLI07Z UME6Pr
Supplier#000007818        uJJOYpaMOri wOStApISY
Supplier#000007885
yXzIOPJJV1Ct76BeZOhgeOqCQQi4K2
Supplier#000007918        A0wREqaUfCn2tHZ
Supplier#000007926        n solT,gR6u
Supplier#000007957        nfvDIDZWzzD6ZrNat1ifscgX5zsP
Supplier#000007965        mT6DVGhTBVKJH
Supplier#000007968        ozlR6G1YlCVj3dRRm
wtLZIGjDnVSYk8k1Bc
Supplier#000007998        gDq8lqL29ldCRNUO0Qzpx5ARfDYb
Supplier#000008168        RcQSYRS1PgpDLnf4
Supplier#000008231        jgTMkwr2HR0 7NB b0wOB4ufp
Supplier#000008243
ZqtMbfGnAEt5sHk8Is3yKlfCSKrmIxOoeucFiik
Supplier#000008275        qFPdnM4K0KC3gaFgd
Supplier#000008323        UViZS
1Eq8wErbcNJM9eOHRyECtMa0qLo3dWpiqP
Supplier#000008366
KTTSOnHZWpy4RcmhFwb75AWIvr89Umqp3dTtM8S
Supplier#000008423        aaJgDHqJ9,oq XivWaZvNeeD,
0KdMrhA
Supplier#000008480        Xz8nqXqo9MHFdHmFV7UP
Supplier#000008532        7OYRAX0Vu5OnclSU61 uK Wu49,
IJm73xJ
Supplier#000008595        fj,IpUXkaXtr64XdrnPoQAEO
Supplier#000008610        9K5KbS,
wbWWYz6d8KsfRtgv3j4qs5Uz5
Supplier#000008705        Rm0y adNbu1WtID8nRcXoMPniC
Supplier#000008742        kEbFansgobnO76f,W0GgB
Supplier#000008841        jrSVfyZzMGQKYu 9isE,
Supplier#000008895        ZOjWCHLQf8277KS
g9PPiauGENlxYm H
Supplier#000008967        ZGvmjuekrTmvCsdjEq6mVEj,
J3yA2OyFhe
Supplier#000008972        wYwlUsnV21dXwIzc3zA5Mfqn7h
Supplier#000009032        Bg0y qU8NtXnsZpa6ldt
Supplier#000009147        Acqx ujuB1SMcwu
Supplier#000009252        mTO1Qc6jvJ,
r1gtIXuEq64oZ3ObrxM,BVE0J
Supplier#000009278        aA27sLuHRXpf3r,FO2LondcMLo
Supplier#000009327        yCAeNMTMTkIRNPaWX
Supplier#000009430        JK9AEEMlyr
Supplier#000009567        AW4LuXOXuznqHOkITZjj2ptC
EFrnRx sy3GwW
Supplier#000009601        WZEUXUPc09wVnDj5l6wfRO9uR
Supplier#000009709        A9DoPk2KnKGRb12Et4g53864,xgK
Supplier#000009753        wfJ5mP9ENTcGhlWmpDkgU1
Supplier#000009796        t0tWiOE ZVocwb6B2k
Supplier#000009799        sWvdH4kQWch4F
Supplier#000009811        nXIxtBT6D1v6TCb2iMYkyU
Supplier#000009812
rbl9euXFoPLlKQVYDVyRouslbbbKDHAkyXY
Supplier#000009862        AhsvM5MdVS
Supplier#000009868        acHkKvVKwyHED66DMttTx
Supplier#000009869        yOfAut Bp7aeMvk9eYRy0Ad7eY,
KZ7yX3KKYEgQK
Supplier#000009899        U3NBqk s Zz06al2m
Supplier#000009974        Uvh0hWngOu96WgB,
OafBQOqwpWqzwg8


204 rows processed.
Statement Processed in 7.30 seconds.

Ended Executing this Stream at Fri Apr  4 11:55:52
2003


Stream Started at 1049486145.08
Stream Ended at 1049486152.38
Stream Processed in 7.30 seconds


SQL statements processed: 1
====================================================
```

qual21.v1

```
====================================================
Begin Execution at Fri Apr  4 11:55:52 2003


-- using default substitutions

select * from (
select
s_name,
count(*)  numwait
from
supplier,
lineitem l1,
orders,
nation
where
s_suppkey = l1.l_suppkey
and o_orderkey = l1.l_orderkey
and o_orderstatus = 'F'
and l1.l_receiptdate > l1.l_commitdate
and exists (
select
*
from
lineitem l2
where
l2.l_orderkey = l1.l_orderkey
and l2.l_suppkey <> l1.l_suppkey
)
and not exists (
select
*
from
lineitem l3
where
l3.l_orderkey = l1.l_orderkey
and l3.l_suppkey <> l1.l_suppkey
and l3.l_receiptdate > l3.l_commitdate
)
and s_nationkey = n_nationkey
and n_name = 'SAUDI ARABIA'
group by
s_name
order by
numwait desc,
s_name)
where rownum <= 100

S_NAME                    NUMWAIT
Supplier#000002829        20.00
Supplier#000005808        18.00
Supplier#000000262        17.00
Supplier#000000496        17.00
Supplier#000002160        17.00
Supplier#000002301        17.00
Supplier#000002540        17.00
Supplier#000003063        17.00
Supplier#000005178        17.00
Supplier#000008331        17.00
Supplier#000002005        16.00
Supplier#000002095        16.00
Supplier#000005799        16.00
Supplier#000005842        16.00
```

```
Supplier#000006450        16.00
Supplier#000006939        16.00
Supplier#000009200        16.00
Supplier#000009727        16.00
Supplier#000000486        15.00
Supplier#000000565        15.00
Supplier#000001046        15.00
Supplier#000001047        15.00
Supplier#000001161        15.00
Supplier#000001336        15.00
Supplier#000001435        15.00
Supplier#000003075        15.00
Supplier#000003335        15.00
Supplier#000005649        15.00
Supplier#000006027        15.00
Supplier#000006795        15.00
Supplier#000006800        15.00
Supplier#000006824        15.00
Supplier#000007131        15.00
Supplier#000007382        15.00
Supplier#000008913        15.00
Supplier#000009787        15.00
Supplier#000000633        14.00
Supplier#000001960        14.00
Supplier#000002323        14.00
Supplier#000002490        14.00
Supplier#000002993        14.00
Supplier#000003101        14.00
Supplier#000004489        14.00
Supplier#000005435        14.00
Supplier#000005583        14.00
Supplier#000005774        14.00
Supplier#000007579        14.00
Supplier#000008180        14.00
Supplier#000008695        14.00
Supplier#000009224        14.00
Supplier#000000357        13.00
Supplier#000000436        13.00
Supplier#000000610        13.00
Supplier#000000788        13.00
Supplier#000000889        13.00
Supplier#000001062        13.00
Supplier#000001498        13.00
Supplier#000002056        13.00
Supplier#000002312        13.00
Supplier#000002344        13.00
Supplier#000002596        13.00
Supplier#000002615        13.00
Supplier#000002978        13.00
Supplier#000003048        13.00
Supplier#000003234        13.00
Supplier#000003727        13.00
Supplier#000003806        13.00
Supplier#000004472        13.00
Supplier#000005236        13.00
Supplier#000005906        13.00
Supplier#000006241        13.00
Supplier#000006326        13.00
Supplier#000006384        13.00
Supplier#000006394        13.00
Supplier#000006624        13.00
Supplier#000006629        13.00
Supplier#000006682        13.00
Supplier#000006737        13.00
Supplier#000006825        13.00
Supplier#000007021        13.00
Supplier#000007417        13.00
Supplier#000007497        13.00
Supplier#000007602        13.00
Supplier#000008134        13.00
Supplier#000008234        13.00
Supplier#000009435        13.00
Supplier#000009436        13.00
Supplier#000009564        13.00
Supplier#000009896        13.00
Supplier#000000379        12.00
Supplier#000000673        12.00
Supplier#000000762        12.00
Supplier#000000811        12.00
Supplier#000000821        12.00
Supplier#000001337        12.00
Supplier#000001916        12.00
Supplier#000001925        12.00
Supplier#000002039        12.00
Supplier#000002357        12.00
Supplier#000002483        12.00

100 rows processed.
Statement Processed in 41.74 seconds.

Ended Executing this Stream at Fri Apr  4 11:56:34
2003


Stream Started at 1049486152.54
Stream Ended at 1049486194.28
Stream Processed in 41.74 seconds


SQL statements processed: 1
==================================================
```

qual22.v1

```
==================================================
Begin Execution at Fri Apr  4 11:56:34 2003


-- using default substitutions

select
cntrycode,
count(*) as numcust,
sum(c_acctbal) as totacctbal
from
(
select
substr(c_phone, 1, 2) as cntrycode,
c_acctbal
from
customer
where
substr(c_phone,1, 2) in
('13', '31', '23', '29', '30', '18', '17')
and c_acctbal > (
select
avg(c_acctbal)
from
customer
where
c_acctbal > 0.00
and substr(c_phone, 1, 2) in
('13', '31', '23', '29', '30', '18', '17')
)
and not exists (
select
*
from
orders
where
o_custkey = c_custkey
)
) custsale
group by
cntrycode
order by
cntrycode

CNTRYCODE NUMCUST               TOTACCTBAL
13        888.00                6737713.99
17        861.00                6460573.72
18        964.00                7236687.40
23        892.00                6701457.95
29        948.00                7158866.63
30        909.00                6808436.13
31        922.00                6806670.18


7 rows processed.
Statement Processed in 6.03 seconds.

Ended Executing this Stream at Fri Apr  4 11:56:40
2003


Stream Started at 1049486194.45
Stream Ended at 1049486200.47
Stream Processed in 6.03 seconds


SQL statements processed: 1
```

Appendix D. Seed and Query Substitution Parameters

This Appendix contains Seed values and substitution parameters for each stream

seed values

```
session 00 326235746
session 01 326235747
session 02 326235748
session 03 326235749
session 04 326235750
session 05 326235751
session 06 326235752
session 07 326235753
session 08 326235754
```

stream 00 substitution parameters

```
14      1995-01-01
2       49      NICKEL  AFRICA
9       cornflower
20      frosted 1993-01-01      IRAQ
6       1995-01-01      0.04    25
17      Brand#35        LG CASE
18      315
8       PERU    AMERICA SMALL ANODIZED STEEL
21      CANADA
13      pending requests
3       MACHINERY       1995-03-05
22      10      19      29      16      23      18
30
16      Brand#53        PROMO ANODIZED 11      14      7
        12      28      49      34      17
4       1993-10-01
11      MOROCCO 0.0000000333
15      1996-08-01
1       86
10      1993-10-01
19      Brand#22        Brand#13        Brand#54        9
        18      25
5       ASIA    1995-01-01
7       INDIA   PERU
12      MAIL    AIR     1994-01-01
```

stream 01 substitution parameters

```
21      SAUDI ARABIA
3       BUILDING        1995-03-21
18      312
5       EUROPE  1995-01-01
11      CANADA  0.0000000333
7       ALGERIA INDONESIA
6       1995-01-01      0.02    24
20      purple  1997-01-01      ARGENTINA
17      Brand#32        LG JAR
12      RAIL    TRUCK   1994-01-01
16      Brand#43        SMALL PLATED    14      36      3
        15      27      30      16      20
15      1994-05-01
13      pending requests
10      1994-07-01
2       37      COPPER  EUROPE
8       INDONESIA       ASIA    STANDARD POLISHED STEEL
14      1995-04-01
19      Brand#24        Brand#51        Brand#54        4
        19      21
9       burlywood
22      11      32      34      27      30      19
33
1       94
4       1996-05-01
```

stream 02 substitution parameters

```
6       1995-01-01      0.07    24
17      Brand#34        LG CAN
14      1995-07-01
16      Brand#23        LARGE POLISHED 29      7
11      27      19      22      8       46
19      Brand#21        Brand#34        Brand#53        9
        20      28
10      1993-04-01
9       bisque
2       24      STEEL   AMERICA
15      1996-11-01
8       ARGENTINA       AMERICA STANDARD BURNISHED STEEL
5       MIDDLE EAST     1995-01-01
22      25      16      21      12      33      22
18
12      AIR     RAIL    1995-01-01
7       PERU    ARGENTINA
13      pending requests
18      314
1       103
4       1994-02-01
20      chiffon 1995-01-01      MOROCCO
3       MACHINERY       1995-03-07
11      MOZAMBIQUE      0.0000000333
21      JORDAN
```

stream 03 substitution parameters

```
8       CHINA   ASIA    PROMO BRUSHED STEEL
5       AFRICA  1996-01-01
4       1996-08-01
6       1996-01-01      0.05    25
17      Brand#31        MED CASE
7       INDONESIA       CHINA
1       111
18      312
22      14      24      26      20      10      27
29
14      1995-11-01
9       yellow
10      1994-01-01
15      1994-08-01
11      EGYPT   0.0000000333
20      mint    1994-01-01      ETHIOPIA
2       12      BRASS   EUROPE
21      ETHIOPIA
19      Brand#33        Brand#22        Brand#42        5
        10      24
13      pending accounts
16      Brand#13        STANDARD ANODIZED       7
16      17      11      24      23      0       21
12      REG AIR RAIL    1995-01-01
3       BUILDING        1995-03-23
```

stream 04 substitution parameters

```
5       AMERICA 1996-01-01
21      RUSSIA
14      1996-02-01
19      Brand#35        Brand#55        Brand#41
10      11      21
15      1997-03-01
17      Brand#32        MED JAR
12      SHIP    RAIL    1996-01-01
6       1996-01-01      0.02    24
4       1994-05-01
9       thistle
8       IRAN    MIDDLE EAST     PROMO PLATED COPPER
16      Brand#43        MEDIUM BURNISHED        25
49      44      3       34      8       21      7
11      PERU    0.0000000333
2       50      NICKEL  AMERICA
10      1994-11-01
18      313
1       119
13      pending accounts
7       ARGENTINA       IRAN
22      30      27      15      24      25      17
11
3       HOUSEHOLD       1995-03-09
20      yellow  1997-01-01      SAUDI ARABIA
```

```
================================================
```

stream 05 substitution parameters

```
================================================
21      KENYA
15      1994-11-01
4       1996-12-01
6       1996-01-01      0.07    24
7       CHINA   BRAZIL
16      Brand#23        ECONOMY POLISHED        27
20      29      1       30      8       32      13
19      Brand#33        Brand#33        Brand#45        5
        12      28
18      315
14      1996-05-01
22      20      25      34      21      17      28
23
11      ETHIOPIA        0.0000000333
13      pending accounts
3       BUILDING        1995-03-25
1       66
2       38      TIN     MIDDLE EAST
5       ASIA    1996-01-01
8       BRAZIL  AMERICA PROMO ANODIZED COPPER
20      indian  1995-01-01      IRAN
12      FOB     TRUCK   1996-01-01
17      Brand#34        MED CAN
10      1993-08-01
9       slate
================================================
```

stream 06 substitution parameters

```
================================================
10      1994-05-01
3       HOUSEHOLD       1995-03-11
15      1997-06-01
13      pending accounts
6       1996-01-01      0.05    25
8       ROMANIA EUROPE  ECONOMY POLISHED COPPER
9       saddle
7       IRAN    ROMANIA
4       1994-09-01
11      CHINA   0.0000000333
22      16      23      31      22      32      20
25
18      312
12      MAIL    TRUCK   1996-01-01
1       74
5       MIDDLE EAST     1996-01-01
16      Brand#13        STANDARD BRUSHED        1
41      13      49      7       21      31      15
2       26      STEEL   AMERICA
14      1996-08-01
19      Brand#45        Brand#25        Brand#35        1
        13      24
20      seashell        1994-01-01      UNITED STATES
17      Brand#31        JUMBO CASE
21      FRANCE
================================================
```

stream 07 substitution parameters

```
================================================
18      314
8       IRAQ    MIDDLE EAST     ECONOMY BURNISHED COPPER
20      deep    1997-01-01      KENYA
21      UNITED KINGDOM
2       13      BRASS   MIDDLE EAST
4       1997-04-01
22      30      15      26      23      19      21
20
17      Brand#33        JUMBO JAR
1       82
11      FRANCE  0.0000000333
9       puff
19      Brand#42        Brand#53        Brand#34        6
        14      20
3       AUTOMOBILE      1995-03-27
13      unusual deposits
5       AFRICA  1996-01-01
7       BRAZIL  IRAQ
10      1993-02-01
16      Brand#43        LARGE BURNISHED         40
31      35      8       32      22      26      19
6       1996-01-01      0.02    24
14      1996-11-01
15      1995-03-01
12      RAIL    TRUCK   1996-01-01
```

```
================================================
```

stream 08 substitution parameters

```
19      Brand#44        Brand#41        Brand#33        1
        15      27
1       90
15      1997-10-01
17      Brand#35        JUMBO CAN
5       AMERICA 1997-01-01
8       CANADA  AMERICA LARGE BRUSHED COPPER
9       papaya
12      AIR     TRUCK   1997-01-01
14      1997-03-01
7       ROMANIA CANADA
4       1995-01-01
3       HOUSEHOLD       1995-03-13
20      pale    1996-01-01      EGYPT
16      Brand#23        PROMO PLATED    37      33
43      40      20      44      8       28
6       1997-01-01      0.08    24
22      14      13      15      20      19      18
12
10      1993-11-01
13      unusual deposits
2       1       NICKEL  ASIA
21      MOZAMBIQUE
18      315
11      ROMANIA 0.0000000333
```

Appendix E. Implementation-Specific Layer/Driver Code

```
=====================================================
```

runTPCHall

```
=====================================================
#!/bin/ksh
. $KIT_DIR/env

ECHO=echo

sqlplus=$ORACLE_HOME/bin/sqlplus
svrmgrl=$ORACLE_HOME/bin/svrmgrl
GTIME=${KIT_DIR}/utils/gtime

RUN_ID_FILE=${KIT_DIR}/audit/r_id

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

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

OUT_DIR=${KIT_DIR}/audit/tests/${RUN_ID}
if [ ! -d $OUT_DIR ]
then
  mkdir $OUT_DIR
fi

SCRIPT_LOG_FILE=${OUT_DIR}/main.out
RDB_TABLES=${OUT_DIR}/rdbtablest
FIRST_TEN=${OUT_DIR}/firstten
LD1DBCRE=${OUT_DIR}/Ld1dbcre
LD2SCTSO=${OUT_DIR}/Ld2sctso
LD3DAPOP=${OUT_DIR}/Ld4dapop
LD4IXCRE=${OUT_DIR}/Ld5ixcre
LD5ANLYZ=${OUT_DIR}/Ld5anlyz

echo Start TPC-H Benchmark SEQUENCE NUMBER: $RUN_ID >
$SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE
echo "Starting a new Oracle log file:
$ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log" >>
$SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE

mv $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log
$ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}
.log.preAudit.$RUN_ID
touch $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log

echo "Start: load database `date`" >> $SCRIPT_LOG_FILE
bumpx.pl -s -x -o 1tera_sas.dat -p dbcre > $LD1DBCRE
bumpx.pl -s -x -o 1tera_sas.dat -p sctso > $LD2SCTSO
STIME=`$GTIME`
echo "Start: timed load portion `date`" >>
$SCRIPT_LOG_FILE
bumpx.pl -s -x -o 1tera_sas.dat -p dapop > $LD3DAPOP
bumpx.pl -s -x -o 1tera_sas.dat -p ixcre > $LD4IXCRE
bumpx.pl -s -x -o 1tera_sas.dat -p anlyz > $LD5ANLYZ
echo "End: timed load portion `date`" >>
$SCRIPT_LOG_FILE

$KIT_DIR/audit/gen_seed.sh $KIT_DIR/audit/seed
echo Generated seed: `cat $KIT_DIR/audit/seed` >>
$SCRIPT_LOG_FILE

/export/home/oracle/local/bin/tshut >>
$SCRIPT_LOG_FILE
sleep 20
/export/home/oracle/local/bin/tstart >>
$SCRIPT_LOG_FILE

echo "Start: dbtables.sql and count.sql" >>
$SCRIPT_LOG_FILE
$sqlplus ${DATABASE_USER} @$KIT_DIR/audit/dbtables >
${RDB_TABLES} 2>&1
$sqlplus ${DATABASE_USER} @$KIT_DIR/audit/firstten >
${FIRST_TEN} 2>&1

echo "End: dbtables.sql and count.sql `date`" >>
$SCRIPT_LOG_FILE


runTPCHpt ${SCALE_FACTOR} 1 ${RUN_ID}

runTPCHpt ${SCALE_FACTOR} 2 ${RUN_ID}

sleep 600
tshut >> $SCRIPT_LOG_FILE

cp $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log
$OUT_DIR

echo "End TPC-H Benchmark SEQUENCE NUMBER: $RUN_ID
`date`" >> $SCRIPT_LOG_FILE

=====================================================
```

runTPCHpt

```
#!/bin/ksh
. $KIT_DIR/env
SCRIPT_DIR=${KIT_DIR}/scripts
SQL_DIR=${KIT_DIR}/sql
UPD_DIR=${KIT_DIR}/update
SRC_DIR=${KIT_DIR}/utils
QRY_DIR=${KIT_DIR}/queries # this is the location of
the query template file
QGEN_DIR=${KIT_DIR}/dbgen
QGEN=${QGEN_DIR}/qgen
QEXEC=${SRC_DIR}
DSS_QUERY=${KIT_DIR}/queries
export DSS_QUERY

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

TPCD_BIN=${KIT_DIR}/audit/bin

GTIME=${SRC_DIR}/gtime
SEED_FILE=${KIT_DIR}/audit/seed

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

# The defaults

QPROG=${QEXEC}/qexec

usage () {

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

while :
do
   case "$1" in
   -u1) shift; U1PROG=$1;;
   -u2) shift; U2PROG=$1;;
   -p) shift; QPROG=$1;;
```

```
   -o) OSTAT=1;;
   -s) SSTAT=1;;
   -h) usage; exit 0;;
   --) shift; break;;
   esac
   shift;
done


if [ "$#" -ne "3" ]
then
  usage
  exit 1
fi

SF=$1
PARA=$2
RUN_ID=$3

OUT_DIR=${KIT_DIR}/audit/tests/${RUN_ID}
if [ ! -d $OUT_DIR ]
then
       mkdir $OUT_DIR
fi

TPCD_LOG=${OUT_DIR}
TPCD_RPT=${OUT_DIR}
OUT=${OUT_DIR}

let UF_SET="($PARA-1)*($NUM_STREAMS+1)+1"
START_SET=1
let STOP_SET=$NUM_STREAMS
let START_SET_UPDATE="($PARA-1)*($NUM_STREAMS+1)+2"
let STOP_SET_UPDATE="$START_SET_UPDATE+$NUM_STREAMS-1"

TPCD_LOG_FILE=${TPCD_LOG}/m${PARA}s0
TPCD_RPT_FILE=${TPCD_RPT}/m${PARA}s0inter
QRY_FILE=${TPCD_RPT}/qtemp.${PARA}s0
QUERY_PARAMETER=${TPCD_LOG}/qp${PARA}.0
SCRIPT_LOG_FILE=${TPCD_LOG}/m${PARA}timing
UF1_LOG=${TPCD_LOG}/m${PARA}s0rf1
UF2_LOG=${TPCD_LOG}/m${PARA}s0rf2
STREAM_COUNT_LOG=${TPCD_LOG}/m${PARA}tstrcnt

echo "TPC-H Test - RUN:${PARA} SEQUENCE:${RUN_ID}
`date`" > $SCRIPT_LOG_FILE
echo "TPC-H Test - RUN:${PARA} SEQUENCE:${RUN_ID}
`date`" > $TPCD_RPT_FILE
echo "Generates query template file with seed: `cat
$SEED_FILE` for stream 0" >> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE

${QGEN} -c -r `cat $SEED_FILE` -p 0 -s ${SF} -l
$QUERY_PARAMETER > ${QRY_FILE}

START=`$GTIME`
echo "Start Power Test - RUN:${PARA}
SEQUENCE:${RUN_ID} Execution Starts $START, `date`" >>
$SCRIPT_LOG_FILE
echo "" >> $SCRIPT_LOG_FILE

# Execute UF1

SDATE=`date`
UF1_START=`$GTIME`
echo "Start UF1  $UF1_START, `date`" >>
$SCRIPT_LOG_FILE

${ECHO} ${UPD_SPT}/runuf1.sh ${UF_SET} >> $UF1_LOG
2>&1
# Execute Query Stream

UF1_END=`${GTIME}`
E1DATE=`date`

UF1_TIME=`echo $UF1_END - $UF1_START | bc`
echo UF1: Execution Time: $UF1_TIME >>
${TPCD_RPT_FILE}
echo Start Time: $UF1_START, $SDATE >>
${TPCD_RPT_FILE}
echo End Time: $UF1_END, $E1DATE >> ${TPCD_RPT_FILE}
echo "" >> ${TPCD_RPT_FILE}

echo "End UF1 $UF1_END, ${E1DATE}" >> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE

echo "Start Query Part `$GTIME`, `date` " >>
$SCRIPT_LOG_FILE

${QPROG} ${DATABASE_USER} q${QRY_FILE}
l${TPCD_LOG_FILE} r${TPCD_RPT_FILE}  > $DF  2>&1

# Execute UF2

UF2_START=`${GTIME}`
E2DATE=`date`

echo "End Query Part `$GTIME`, ${E2DATE}" >>
$SCRIPT_LOG_FILE
echo "" >> $SCRIPT_LOG_FILE

echo "Start UF2  $UF2_START, `date`" >>
$SCRIPT_LOG_FILE
${ECHO} ${UPD_SPT}/runuf2.sh ${UF_SET} >> $UF2_LOG
2>&1
UF2_END=`${GTIME}`
END=`${GTIME}`
EDATE=`date`

echo "End UF2 $UF2_END, $EDATE" >>  $SCRIPT_LOG_FILE
echo >>  $SCRIPT_LOG_FILE

echo "End TPC-H Power Test - RUN:${PARA}
SEQUENCE:${RUN_ID}, $END, $EDATE" >> $SCRIPT_LOG_FILE
MEA_INT=`echo $END - $START | bc`
echo "Elapsed Time for TPC-H Power Test - RUN:${PARA}
SEQUENCE:${RUN_ID} is $MEA_INT" >> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE

UF2_TIME=`echo $UF2_END - $UF2_START | bc`
echo UF2: Execution Time: $UF2_TIME >>
${TPCD_RPT_FILE}echo Start Time: $UF2_START, $E2DATE
>> ${TPCD_RPT_FILE}
echo End Time: $UF2_END, $EDATE >> ${TPCD_RPT_FILE}

${KIT_DIR}/audit/abridge.pl ${TPCD_LOG_FILE}

i=$START_SET
PSEED=`cat $SEED_FILE`

while [ $i -le $STOP_SET ]; do
  TPCD_LOG_FILE=${TPCD_LOG}/mt${RUN_ID}_${i}.log
  TPCD_RPT_FILE=${TPCD_RPT}/mt${RUN_ID}_${i}.rpt
        QUERY_PARAMETER=${TPCD_LOG}/qp${PARA}.${i}
  QRY_FILE=${TPCD_RPT}/qtemp.${PARA}s${i}

  PSEED=`expr $PSEED + 1`
  ${QGEN} -c -r ${PSEED} -p ${i} -s ${SF} -l
$QUERY_PARAMETER > ${QRY_FILE}

  i=`expr $i + 1`
done

TH_START_D=`date`
TH_START_T=`${GTIME}`
echo >> $SCRIPT_LOG_FILE

rm -f /tmp/th_pipe1
mknod /tmp/th_pipe1 p
rm -f /tmp/th_pipe2
mknod /tmp/th_pipe2 p
i=$START_SET

echo "Start Throughput Test - RUN:${PARA}
SEQUENCE:${RUN_ID} $TH_START_T, $TH_START_D"  >>
$SCRIPT_LOG_FILE

# starts a script to count the streams during the
throughtput run
(scnt.sh $PARA $RUN_ID > $STREAM_COUNT_LOG &)
scnt_PID=$!

while [ $i -le $STOP_SET ]; do
  M_SDATE=`date`
  M_STIME=`${GTIME}`
        TPCD_LOG_FILE=${TPCD_LOG}/m${PARA}s${i}
        TPCD_RPT_FILE=${TPCD_RPT}/m${PARA}s${i}inter
  echo "Start Query Stream $i $M_STIME, ${M_SDATE}" >>
$SCRIPT_LOG_FILE
        QRY_FILE=${TPCD_RPT}/qtemp.${PARA}s${i}
  ${QPROG} ${DATABASE_USER} q${QRY_FILE}
l${TPCD_LOG_FILE} r${TPCD_RPT_FILE} | grep -v
```

```
"Connected to ORACLE" >> $SCRIPT_LOG_FILE &
  i=`expr $i + 1`
done

(${KIT_DIR}/audit/runTPCHus $RUN_ID $START_SET_UPDATE
$STOP_SET_UPDATE ${SF} $PARA >> $SCRIPT_LOG_FILE 2>&1
&)

wait
THQ_END_T=`$GTIME`
THQ_END_D=`date`
echo End all Query Streams $THQ_END_T, $THQ_END_D >>
$SCRIPT_LOG_FILE
print > /tmp/th_pipe1
read < /tmp/th_pipe2

TH_END_D=`date`
TH_END_T=`$GTIME`
echo End Update Stream ${TH_END_T}, ${TH_END_D} >>
$SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE
echo "End Throughput Test ${TH_END_T}, ${TH_END_D}" >>
$SCRIPT_LOG_FILE
echo Execution Time Throughput Test: `echo ${TH_END_T}
- ${TH_START_T} | bc` >> $SCRIPT_LOG_FILE

i=$START_SET
while [ $i -le $STOP_SET ]; do
        TPCD_LOG_FILE=${TPCD_LOG}/m${PARA}s${i}
  ${KIT_DIR}/audit/abridge.pl ${TPCD_LOG_FILE}
        i=`expr $i + 1`
done
#kill -9 $scnt_PID
/usr/bin/pkill scnt
```

==========

runTPCHus

==========

```
#!/bin/ksh
. $KIT_DIR/env

SCRIPT_DIR=${KIT_DIR}/scripts
SQL_DIR=${KIT_DIR}/sql
UPD_DIR=${KIT_DIR}/update
UPD_SPT=${UPD_DIR}/scripts
SRC_DIR=${KIT_DIR}/utils
QRY_DIR=${KIT_DIR}/queries # this is the location of
the query template file
QGEN_DIR=${KIT_DIR}/dbgen
QGEN=${QGEN_DIR}/qgen

DSS_QUERY=${KIT_DIR}/queries
export DSS_QUERY

RUN_ID=$1
START_SET_UPDATE=$2
STOP_SET_UPDATE=$3
SF=$4
PARA=$5

OUT_DIR=${KIT_DIR}/audit/tests/${RUN_ID}
if [ ! -d $OUT_DIR ]
then
        mkdir $OUT_DIR
fi


TPCD_RPT=$OUT_DIR
SCRIPT_LOG_FILE=${OUT_DIR}/m${PARA}timing
OUT=$OUT_DIR

GTIME=${SRC_DIR}/gtime
HID=1

START=`$GTIME`
echo "Start Update Stream $START, `date`" >>
$SCRIPT_LOG_FILE
echo "" >> $SCRIPT_LOG_FILE


#waiting for all the query streams to finish first
read < /tmp/th_pipe1

i=$START_SET_UPDATE
j=1
while [ $i -le $STOP_SET_UPDATE ]; do

  # Execute UF1

  UF1_LOG=${OUT_DIR}/m${PARA}s${j}rf1
  UF2_LOG=${OUT_DIR}/m${PARA}s${j}rf2
  RPT_FILE=${OUT_DIR}/m${PARA}s${j}inter

  SDATE=`date`
  UF1_START=`$GTIME`
        echo "Start UF1-${j} at ${UF1_START}, ${SDATE}"
>> ${RPT_FILE}

  ${UPD_SPT}/runuf1.sh ${i} >> ${UF1_LOG} 2>&1
  UF1_END=`${GTIME}`
        EDATE=`date`
  echo "End UF1-${j} at ${UF1_END}, ${EDATE}" >>
${RPT_FILE}
        echo UF1-${j} Execution Time: `echo ${UF1_END}
- ${UF1_START} | bc` >> ${RPT_FILE}

  # Execute UF2

  SDATE=`date`
  UF2_START=`${GTIME}`
  echo "Start UF2-${j} ${UF2_START}, ${SDATE}" >>
${RPT_FILE}

  ${UPD_SPT}/runuf2.sh ${i} >> ${UF2_LOG} 2>&1
  UF2_END=`${GTIME}`
  EDATE=`date`
  echo "End UF2-${j} at $UF2_END, ${EDATE}" >>
${RPT_FILE}
        echo UF2-${j} Execution Time: `echo ${UF2_END}
- ${UF2_START} | bc` >> ${RPT_FILE}

  i=`expr $i + 1`
  j=`expr $j + 1`
done

print > /tmp/th_pipe2
```

==========

runuf1.sh

==========

```
#!/bin/ksh
. $KIT_DIR/env
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
LOG_DIR=${UPDATE_DIR}/log
GTIME=${SCRIPT_DIR}/gtime
SF=${SCALE_FACTOR}
PAR_HINT=${UPDATE_DOP_INS}

LOGPATH=.
PASSWD=${DATABASE_USER}

if [ $# -lt 1 ];
then
   echo runuf1.sh setnum
   exit 1
fi
SETNUM=$1
i=1
PID=""

# perform the update function 1

START=`$GTIME`

# first create the temp tables

sqlplus /NOLOG << !

connect $PASSWD;
set timing  on
set serveroutput on
set echo on

drop directory data_dir;
create directory data_dir as '${KIT_DIR}
/update/update_sets';

drop table temp_l_et;
create table temp_l_et(
```

```
    l_shipdate          date ,
    l_orderkey          number ,
    l_discount          number ,
    l_extendedprice     number ,
    l_suppkey           number ,
    l_quantity          number ,
    l_returnflag        char(1) ,
    l_partkey           number ,
    l_linestatus        char(1) ,
    l_tax               number ,
    l_commitdate        date ,
    l_receiptdate       date ,
    l_shipmode          char(10) ,
    l_linenumber        number ,
    l_shipinstruct      char(25) ,
    l_comment           varchar(44)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                records delimited by newline
                badfile 'l_et.${SETNUM}.bad'
                logfile 'l_et.${SETNUM}.log'
                fields terminated by '|'
                missing field values are null
        )
        location (
        'lineitem.tbl.u${SETNUM}'))
reject limit unlimited;

drop table temp_o_et;
create table temp_o_et(
    o_orderdate         date ,
    o_orderkey          number ,
    o_custkey           number ,
    o_orderpriority     char(15) ,
    o_shippriority      number ,
    o_clerk             char(15) ,
    o_orderstatus       char(1) ,
    o_totalprice        number ,
    o_comment           varchar(79)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                records delimited by newline
                badfile 'o_et.${SETNUM}.bad'
                logfile 'o_et.${SETNUM}.log'
                fields terminated by '|'
                missing field values are null
        )
        location (
        'orders.tbl.u${SETNUM}'))
reject limit unlimited;

alter table temp_l_et parallel  ${PAR_HINT};
alter table temp_o_et parallel  ${PAR_HINT};


alter session force parallel dml parallel (degree
${PAR_HINT});
alter session set isolation_level = serializable;



insert into orders (select * from temp_o_et);

insert into lineitem (select * from temp_l_et);

commit;

drop table temp_l_et;
drop table temp_o_et;

exit;
!

END=`$GTIME`

# Done

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

==========

runuf2.sh

==========

```
#!/bin/ksh
. $KIT_DIR/env
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
GTIME=${SCRIPT_DIR}/gtime
LOG_DIR=${UPDATE_DIR}/log
PAR_HINT=${UPDATE_DOP_DEL}
SF=${SCALE_FACTOR}
PASSWD=${DATABASE_USER}

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

SETNUM=$1

i=1
PID=""

START=`$GTIME`
# first create the temp tables

sqlplus /NOLOG << !

connect $PASSWD;
set timing  on
set serveroutput on
set echo on

drop directory data_dir;
create directory data_dir as '${KIT_DIR}
/update/update_sets';

drop table temp_okey_et;
drop table temp_okey;

create table temp_okey_et(
    t_orderkey          number
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                records delimited by newline
                badfile 'okey.${SETNUM}.bad'
                logfile 'okey.${SETNUM}.log'
                fields terminated by '|'
                missing field values are null
        )
        location (
        'delete.${SETNUM}'))
reject limit unlimited;

alter table temp_okey_et parallel 16;

create table temp_okey parallel  16  nologging as
select * from temp_okey_et;

create unique index i_temp_okey on temp_okey
(t_orderkey) parallel 16 nologging compute statistics;
analyze table temp_okey estimate statistics sample 2
percent;

alter session  force parallel dml parallel
${PAR_HINT};
alter session set isolation_level=serializable;



delete from (select /*+ index(o) use_nl(o) */ o.rowid
from orders o, temp_okey t where o.o_orderkey =
t.t_orderkey order by 1);

delete from (select /*+ index(l) use_nl(l) */ l.rowid
from lineitem l,temp_okey t where l.l_orderkey =
t.t_orderkey order by 1);
```

```
commit;

drop table temp_okey;
drop table temp_okey_et;
exit;
!

END=`$GTIME`

# Done

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

==========

env

==========

```
export KIT_DIR=/dbms/oracle9i/kit
export DSS_QUERY=/dbms/oracle9i/kit/queries
export DSS_CONFIG=/dbms/oracle9i/kit/dbgen
```

==========

qexecpl.c

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

#include "qexecpl.h"

/* Function Prototypes */

extern double gettime();

/* function prototypes from gen.c */

int get_statement();

/* Declare error handling functions */

void sql_error();

/* Other prototypes */

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

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

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

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

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

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

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

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

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

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

char stmt[SQL_LEN];       /* The SQL statement or
comment line.  */
char qn[3];               /* Number of the query being
executed  */
char qnp[3];              /* Number of the previous
query executed  */
char cmnt[5000];            /* Buffer to save the
comment.         */
#ifdef LINUX
FILE *qtemp;      /* fd for query template
*/
FILE *logfile;   /* log and report files
*/
FILE *rep;
#else
FILE *qtemp = stdin;      /* fd for query template
*/
FILE *logfile = stdout;   /* log and report files
*/
FILE *rep = stdout;
#endif
void *defbuf;             /* Buffer pointer for ODEFIN
*/
int deflen = 0;           /* Size of data type for
ODEFIN        */
int deftype = 1;          /* Oracle type number for
ODEFIN       */

int pfmem = PFMEMSIZE;    /* Memory to prefetch rows
*/

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

/* OCI handles */

OCIEnv *tpcenv = NULL;
OCIServer *tpcsrv = NULL;
OCIError *errhp = NULL;
OCISvcCtx *tpcsvc = NULL;
OCISession *tpcusr = NULL;
OCIStmt *curq = NULL;
OCIStmt *cur_dml = NULL;
OCIStmt *cur_ddl = NULL;
OCIParam *tpcpar = NULL;

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

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

void usage() {

  fprintf(stderr,"\nUsage: qexec username/password
[q<path name for query template file>]\n");
```

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



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

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

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

  /* Rollback just in case */

  (void) OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);

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

  SQLexit();

  exit(1);
}

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

  int i,pos,pos2;
  int retcode;      /* Return code for get_statement
*/
#ifdef LINUX
  logfile=fopen("/dev/stdout","w");
  qtemp=fopen("/dev/stdin","rw");
  rep=fopen("/dev/stdout","w");
#endif
  /* Initialize some variables */

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

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

  strcpy(logname, argv[1]);

  /* Process  optional parameters */

  argc -= 1;
  argv += 1;

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

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

  SQLinit();

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

  time(&tim);
  fprintf(logfile, "Begin Execution at %s\n\n",
ctime(&tim));
  fprintf(rep, "Begin Executing this Stream at %
s\n\n", ctime(&tim));
  /* Get the next statement and start processing it */

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

```
    switch (retcode) {

      /* If this is a comment, skips it */
    case COMMENT:
      /*if (end_flag) {
        end_flag = 0;  */    /* reset query end flag */
       /* save the comment so that we can print it out
later on  */
        /* strcpy(cmnt, stmt);
        break;
      } */
      if (stmt[3]== '@') {
         pos=4;
         strcpy(qnp,qn);
         while (stmt[pos] != ')') {
           pos++;
         }
         pos2=0;
         pos++;
         while (stmt[pos] != '.') {
           /*printf ("qn %d %c \n",pos2,stmt[pos]);*/
           qn[pos2]=stmt[pos];
           pos2++;
           pos++;
         }
         qn[pos2] = 0;
         /* printf("found a new query: %s\n",qn); */
      }
      /* save the comment so that we can print it out
later on  */
      strcat(cmnt, stmt);
      break;

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

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

      /* Executes the query */
      SQLexec();

      stmt_cnt++;
      qry_cnt++;
      fflush(rep);
      fflush(logfile);
      /*
      fprintf(logfile,"\nStatement Started at %.2f\n",
s_tr_start);
      fprintf(logfile,"Statement Ended at %.2f\n",
s_tr_end);

      fprintf(logfile,"Statement Processed in %.2f
seconds.\n",
              (s_tr_end - s_tr_start));
      fprintf(rep, "Query %s: Execution Time: %.2f
started %.2f ended %.2f\n",
                    qn,(s_tr_end -
s_tr_start)s_tr_start,s_tr_end);
      fflush(rep);
      fflush(logfile);*/
      break;

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

  /* Get Timing for the last query */

  tr_end = gettime();

  fprintf(logfile,"Query Processed in %.2f
seconds.\n\n",(tr_end - s_tr_start));

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

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

  /* fprintf(rep, "Query %s : Execution time %.2f\n",
qn,(tr_end - s_tr_start));*/
  fprintf(rep, "Query %s: Execution Time: %.2f started
%.2f ended %.2f\n",
                     qn,(tr_end - s_tr_start),
s_tr_start,tr_end);

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

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

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

  fflush(rep);
  fflush(logfile);

  /* Close the query template file */

  fclose(qtemp);

  /*  Disconnect from ORACLE. */

  SQLexit();
  exit(0);
}


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

void SQLinit() {

  int i;

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

  for (i=0; i<MAX_SEL_LIST; i++) {
    if (i < MAX_PREALLOC) {
      dlist[i] = (dltype *) memalloc (sizeof(dltype));
      dlist[i]->defhdl = NULL;
/*      OCIhalloc(curq,&(dlist[i]->defhdl),
OCI_HTYPE_DEFINE); */
    }
    else
      dlist[i] = NULL;
  }

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

  (void) OCIInitialize(OCI_DEFAULT,(dvoid *)0,0,0,0);

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

  OCIhalloc(tpcenv,&errhp,OCI_HTYPE_ERROR);
  OCIhalloc(tpcenv,&curq,OCI_HTYPE_STMT);
  OCIhalloc(tpcenv,&cur_dml,OCI_HTYPE_STMT);
  OCIhalloc(tpcenv,&cur_ddl,OCI_HTYPE_STMT);
```

```
  OCIhalloc(tpcenv,&tpcsvc,OCI_HTYPE_SVCCTX);
  OCIhalloc(tpcenv,&tpcsrv,OCI_HTYPE_SERVER);
  OCIhalloc(tpcenv,&tpcusr,OCI_HTYPE_SESSION);

  /* get username and password */

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

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

  OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcsrv,0,
OCI_ATTR_SERVER,errhp);
  OCIaset(tpcusr,OCI_HTYPE_SESSION,logname,
strlen(logname),OCI_ATTR_USERNAME,
          errhp);
  OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,
strlen(passwd),OCI_ATTR_PASSWORD,
          errhp);

  if ((status = OCISessionBegin(tpcsvc, errhp, tpcusr,
OCI_CRED_RDBMS,
                                OCI_DEFAULT)) !=
OCI_SUCCESS)
    sql_error(errhp,status,1);

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

/*
  if ((status=OCILogon((OCIEnv *)tpcenv,(OCIError
*)errhp,(OCISvcCtx *)tpcsvc,
                (text *)logname, strlen(logname),
(text *)passwd,
                strlen(passwd), (text *) 0, 0)) !=
OCI_SUCCESS)
    sql_error(errhp, status, 1);
*/
  printf("\nConnected to ORACLE as user: %s\n\n",
logname);

}



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

void SQLexec()
{
  int i;
  ub2 stmttyp = OCI_STMT_SELECT;     /* default is a
SELECT statement */

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

  if (qry_cnt) {
     time(&tim);
     s_tr_end = gettime();
     fprintf(logfile,"Query Processed in %.2f
seconds.\n\n",
      (s_tr_end - s_tr_start));

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

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

     /*fprintf(rep, "Query %s : Execution time %
.2f\n", qnp,(s_tr_end - s_tr_start));*/
     fprintf(rep, "Query %s: Execution Time: %.2f
started %.2f ended %.2f\n",
                      qnp,(s_tr_end - s_tr_start),
s_tr_start,s_tr_end);

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

     fflush(logfile);
     fflush(rep);
  }
  else
     tr_start = gettime();

  s_tr_start = gettime();

  /* prepare the statement */

  if ((status = OCIStmtPrepare(curq, errhp, (text*)
stmt, (ub4) strlen(stmt),
                               OCI_NTV_SYNTAX,
OCI_DEFAULT)) != OCI_SUCCESS)
    sql_error(errhp,status,1);

  /* Prints the query text and comment to the logfile
*/

  fprintf(logfile, "\n%s\n", cmnt);
  cmnt[0]=0;
  fprintf(logfile, "\n%s\n", stmt);

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

  OCIaget(curq,OCI_HTYPE_STMT,&stmttyp,NULL,
OCI_ATTR_STMT_TYPE,errhp);

  if (stmttyp != OCI_STMT_SELECT) {
    OCIsexec(tpcsvc,curq,errhp,1);
    return;
  }

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

  /* first let's execute it to get the select-list
definition */

  OCIaset(curq, OCI_HTYPE_STMT, &pfmem, 0,
OCI_ATTR_PREFETCH_MEMORY, errhp);

  OCIsexec(tpcsvc,curq,errhp,0);

  num_sel_list = define_output_variables();

  /* Executes the query and fetches the rows */

  (void) process_select_list(num_sel_list);

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

  OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,
OCI_ATTR_ROW_COUNT,errhp);

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

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

  /* reset set_fetchrows */

  num_to_fetch = -1;

}
```

```
void SQLexit() {

  int i;

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

  /* free all memory */

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

  /* Flush all output */

  fflush(rep);
  fflush(logfile);

}


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

int define_output_variables()
{

  int i;
  int retflag = 0;

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

    slist[i].buflen = MAX_COLNAME_SIZE;

    if (OCIParamGet(curq, OCI_HTYPE_STMT, errhp,
(dvoid **) &tpcpar,
                            POS(i)) != OCI_SUCCESS)
      break;


    /* dsize and nullok fields of dlist not used */

    OCIaget(tpcpar, OCI_DTYPE_PARAM,
&(slist[i].dbsize),
            NULL, OCI_ATTR_DATA_SIZE, errhp);
    OCIaget(tpcpar, OCI_DTYPE_PARAM,
&(slist[i].dbtype),
            NULL, OCI_ATTR_DATA_TYPE, errhp);
    OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].buf),
            &(slist[i].buflen), OCI_ATTR_NAME, errhp);
    OCIaget(tpcpar, OCI_DTYPE_PARAM,
&(slist[i].precision),
            NULL, OCI_ATTR_PRECISION, errhp);
    OCIaget(tpcpar, OCI_DTYPE_PARAM,
&(slist[i].scale),
            NULL, OCI_ATTR_SCALE, errhp);

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

/*
    if (slist[i].buflen < MAX_COLNAME_SIZE)
      (slist[i].buf)[slist[i].buflen] = '\0';
*/
    /* Well, we need to allocate for entries for dlist
*/

    if (i >= MAX_PREALLOC) {
      dlist[i] = (dltype *) memalloc(sizeof(dltype));
      dlist[i]->defhdl = NULL;
    }

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

    switch (slist[i].dbtype) {

    case OCI_TYPECODE_NUMBER:

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

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

      break;

    default:

      /* default is character string */

      defbuf = (char **) dlist[i]->sbuf;
      deflen = MAX_STR_LEN;
      deftype = SQLT_STR;
/*      deftype = OCI_TYPECODE_CHAR; */
      break;
    }

    /* Define the column */

    if ((status=OCIDefineByPos(curq,&(dlist[i]-
>defhdl),errhp,POS(i),
                               defbuf,deflen,deftype,
NULL,
                               dlist[i]->rlen,NULL,
OCI_DEFAULT))!=OCI_SUCCESS)
      sql_error(errhp,status,1);
  }
  return i;
}


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

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

  int i,j;
  int ntf;
  int num_so_far;
  sword stats = OCI_SUCCESS;

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

  print_header(num);

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

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

  /* Fetch the rows and print them out */

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

    stats = OCIStmtFetch(curq, errhp, MAX_ARRAY,
OCI_FETCH_NEXT, OCI_DEFAULT);
```

```
    OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,
OCI_ATTR_ROW_COUNT,errhp);

    print_rows(num,rows_ret);

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

    if (stats != OCI_NO_DATA) {
      if (num_to_fetch == -1) {
        while ((stats = OCIStmtFetch(curq,errhp,
MAX_ARRAY,OCI_FETCH_NEXT,
                                     OCI_DEFAULT)) ==
OCI_SUCCESS) {
          OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                  OCI_ATTR_ROW_COUNT,errhp);
          print_rows(num,(num_so_far-rows_ret));
          rows_ret = num_so_far;
        }
        /* Print the final rows */
        OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                OCI_ATTR_ROW_COUNT,errhp);
        print_rows(num,(num_so_far-rows_ret));
        rows_ret = num_so_far;
      } else {
        ntf -= MAX_ARRAY;

        while ((stats = OCIStmtFetch(curq,errhp,
                                     ((ntf>MAX_ARRAY) ?
MAX_ARRAY:ntf),
                                     OCI_FETCH_NEXT,
OCI_DEFAULT)) ==
                OCI_SUCCESS) {
          ntf -= MAX_ARRAY;
          OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                  OCI_ATTR_ROW_COUNT,errhp);
          print_rows(num,(num_so_far-rows_ret));
          rows_ret = num_so_far;
          if (ntf <= 0) break;
        }
        OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                OCI_ATTR_ROW_COUNT,errhp);
        print_rows(num,(num_so_far-rows_ret));
        rows_ret = num_so_far;
      }
    }
  } else {
    OCIStmtFetch(curq, errhp, ntf, OCI_FETCH_NEXT,
OCI_DEFAULT);
    OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,
OCI_ATTR_ROW_COUNT,errhp);
    print_rows(num,rows_ret);
  }

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

}


int get_statement()
{

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

  /* Reset statement buffer */

  stmt[0] = '\0';

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

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

    /* remove blanks */

    str = line;

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

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

    strcat(stmt, str);

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

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

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


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

void *memalloc(size)
     int size;
{

  void *tmp;

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



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

  int i, diff;
  char colname[MAX_COLNAME_SIZE];
  int len = 0;           /* Running column length */
  int cwid = 0;

  fprintf(logfile, "\n");

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

    /* extract the column name */

    strncpy((char *)colname, (char *)slist[i].buf,
slist[i].buflen);
    colname[slist[i].buflen] = '\0';

    /* format the output a little */

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

    /* do a little bit of formatting */

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

```
  }

  fprintf(logfile,"\n");
}


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

  int i,j;
  int len;
  int diff;
  int cwid;

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

    len = 0;

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

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

      /* do a little bit of formatting */

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

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


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

void remove_newline(str)
     char *str;
{

  char *p;

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

==========

qexecpl.h

```
#ifndef QSTREAMPL_H

#define QSTREAMPL_H

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

#include <oratypes.h>

#include <oratypes.h>

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

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

/* some basic definitions */

#define UNAME_LEN 64
#define MAX_FILE_PATH_LEN 128

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

#ifndef FALSE
#define FALSE 1
#endif /* FALSE */
#ifndef LINUX
#define MAX(x,y) ((x >= y) ? x : y)
#define MIN(x,y) ((x <= y) ? x : y)
#endif
/* defines and typedefs for parsing */

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

/* defines and typedefs for query description */

#define MAX_COLNAME_SIZE 32   /* Maximum length of
Column name  */
#define MAX_SEL_LIST 16       /* Maximum items on a
select list */

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

/* types for describe */

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

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

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

typedef struct des
{
  ub2 dbsize;
  ub4 buflen;
/*  sb2 dsize;  */
  sb4 scale;
/*  sb2 nullok; */
  OCITypeCode dbtype;
/*  text buf[MAX_COLNAME_SIZE]; */
```

```
  text *buf;
  ub1 precision;
} sltype;


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

#define MAX_ARRAY 50       /* Maximum array size for
array fetch */
#define PFMEMSIZE 65536    /* Memory size of prefetch
buffer     */

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

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

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

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

extern int errno;

#define SQL_LEN 2048

#ifndef NULL
#define NULL 0
#endif

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

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#ifndef ub1
#define ub1 unsigned char
#endif

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

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

/* For get_statement */

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


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

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

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

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

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


#define ISOTXT "alter session set isolation_level =
serializable"
#define PDMLTXT "alter session force parallel dml
parallel (degree 84)"
#define PDDLTXT "alter session force parallel ddl
parallel (degree 84)"

#endif /* QSTREAMPL_H */
```

==========

gtime.c

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

# include <sys/time.h>

main ()
{

  struct timeval tv;

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

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

}

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

Appendix F. Misc database scripts

```
Activity Between Database Load and Run1 When the load
finished, the runTPCHall script automatically selected
a seed value and saved it.

The database was restarted.

Then the 2 auditor scripts count.sql and dbtables.sql
were run to validate that the database structure was
correct.


==================================================
```

count.sql

```
==================================================
select * from lineitem where rownum < 11;
select * from orders where rownum < 11;
select * from part where rownum < 11;
select * from partsupp where rownum < 11;
select * from supplier where rownum < 11;
select * from customer where rownum < 11;
select * from nation where rownum < 11;
select * from region where rownum < 11;

==================================================
```

dbtables.sql

```
==================================================
set numwidth 25
SELECT COUNT(*) FROM LINEITEM;

SELECT * FROM LINEITEM
WHERE L_ORDERKEY IN
 ( 4, 26598,  148577,  387431, 56704, 517442, 600000)
AND L_LINENUMBER = 1
ORDER BY L_ORDERKEY;



SELECT * FROM REGION;

SELECT COUNT(*) FROM NATION;

SELECT * FROM NATION
WHERE N_NATIONKEY IN (3,10,14,20)
ORDER BY N_NATIONKEY;



SELECT COUNT(*) FROM ORDERS;

SELECT * FROM ORDERS
WHERE O_ORDERKEY IN ( 7, 44065, 287590, 411111,
483876, 599942 )
ORDER BY O_ORDERKEY;



SELECT COUNT(*) FROM PART;

SELECT * FROM PART
WHERE P_PARTKEY IN (1,984,8743,9028,13876,17899,20000)
ORDER BY P_PARTKEY;



SELECT COUNT(*) FROM PARTSUPP;

SELECT* FROM PARTSUPP
   WHERE PS_PARTKEY = 3398
   AND PS_SUPPKEY = (SELECT MIN(PS_SUPPKEY)
      FROM PARTSUPP WHERE PS_PARTKEY = 3398);

SELECT* FROM PARTSUPP
   WHERE PS_PARTKEY =15873
   AND PS_SUPPKEY = (SELECT MIN(PS_SUPPKEY)
      FROM PARTSUPP WHERE PS_PARTKEY = 15873);

SELECT* FROM PARTSUPP
   WHERE PS_PARTKEY = 11394
   AND PS_SUPPKEY = (SELECT MIN(PS_SUPPKEY)
      FROM PARTSUPP WHERE PS_PARTKEY = 11394);

SELECT* FROM PARTSUPP
   WHERE PS_PARTKEY = 6743
   AND PS_SUPPKEY = (SELECT MIN(PS_SUPPKEY)
      FROM PARTSUPP WHERE PS_PARTKEY = 6743);

SELECT* FROM PARTSUPP
   WHERE PS_PARTKEY = 19763
   AND PS_SUPPKEY = (SELECT MIN(PS_SUPPKEY)
      FROM PARTSUPP WHERE PS_PARTKEY =19763);



SELECT COUNT(*) FROM SUPPLIER;

SELECT * FROM SUPPLIER
WHERE  S_SUPPKEY IN (83,265,492,784,901,1000)
ORDER BY S_SUPPKEY;



DROP TABLE MINMAX;

CREATE TABLE MINMAX
(TNAME CHAR(15),
KEYMIN INTEGER,
KEYMAX INTEGER);

INSERT INTO MINMAX
SELECT 'LINEITEM_ORD',MIN(L_ORDERKEY),MAX(L_ORDERKEY)
FROM LINEITEM ;

INSERT INTO MINMAX
SELECT 'LINEITEM_NBR',MIN(L_LINENUMBER),
MAX(L_LINENUMBER)
FROM LINEITEM;

INSERT INTO MINMAX
SELECT 'ORDERTBL',MIN(O_ORDERKEY),MAX(O_ORDERKEY)
FROM ORDERS;

INSERT INTO MINMAX
SELECT 'CUSTOMER',MIN(C_CUSTKEY),MAX(C_CUSTKEY)
FROM CUSTOMER;

INSERT INTO MINMAX
SELECT 'PART',MIN(P_PARTKEY),MAX(P_PARTKEY)
FROM PART;

INSERT INTO MINMAX
SELECT 'SUPPLIER',MIN(S_SUPPKEY),MAX(S_SUPPKEY)
FROM SUPPLIER;

INSERT INTO MINMAX
SELECT 'PARTSUPP_PART',MIN(PS_PARTKEY),MAX(PS_PARTKEY)
FROM PARTSUPP;

INSERT INTO MINMAX
SELECT 'PARTSUPP_SUPP',MIN(PS_SUPPKEY),MAX(PS_SUPPKEY)
FROM PARTSUPP ;

INSERT INTO MINMAX
SELECT 'NATION',MIN(N_NATIONKEY),MAX(N_NATIONKEY)
FROM NATION;

INSERT INTO MINMAX
SELECT 'REGION',MIN(R_REGIONKEY),MAX(R_REGIONKEY)
FROM REGION;

SELECT * FROM MINMAX;

==================================================
```

tshut

```
==================================================
#!/bin/ksh

if [ "$2" != "" -a "$2" != "1" ]; then
  INUM=$2
  if [ -f $ORACLE_HOME/work/t_init$INUM.ora ]; then
    export ORACLE_SID="$ORACLE_SID"$INUM
  fi
fi

if [ "$1" = "abort" ]; then
svrmgrl << !
connect internal
```

```
shutdown abort
exit
!
else
svrmgrl << !
connect internal
shutdown immediate
exit
!
fi
```

==========

tstart

==========

```
#!/bin/ksh

if [ $# -ne 0 ]; then
  INUM=$1
else
  INUM=1
fi

echo
echo "Starting instance $INUM"

if [ -f $ORACLE_HOME/dbs/init_"$ORACLE_SID".ora ];
then
  PFILE=$ORACLE_HOME/dbs/init_"$ORACLE_SID".ora
else
  if [ -f $ORACLE_HOME/work/t_init$INUM.ora ]; then
    if [ $INUM -ne 1 ]; then
      export ORACLE_SID="$ORACLE_SID"$INUM
    fi
    PFILE=$ORACLE_HOME/work/t_init$INUM.ora
  else
    if [ -f $ORACLE_HOME/dbs/tkinit.ora ]; then
      PFILE=$ORACLE_HOME/dbs/tkinit.ora
    else
      echo "What pfile should I use ???"
      exit 1
    fi
  fi
fi


sqlplus /NOLOG << !
connect / as sysdba
startup pfile=$PFILE

alter system switch logfile;
alter system switch logfile;
alter system switch logfile;

exit
!
```

Appendix G. Pricing information

For Oracle pricing please contact:

MaryBeth Pierantoni
650-506-2118
mary.beth.pierantoni@oracle.com

For Sun pricing please contact:

Daryl Madura
503-617-8588
daryl.madura@sun.com