Hewlett-Packard Company

TPC Benchmark™ H
Full Disclosure Report

HP BladeSystem ProLiant BL25p Cluster 8P
using
Oracle Database 10g Enterprise Edition with
Real Application Cluster and
Partitioning; and
Red Hat Enterprise Linux 4 ES

First Edition
September 2005

First Edition – September 2005

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

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

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

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

ORACLE 10g, SQL*DBA, SQL*Loader, SQL*Net, SQL*Plus, Pro *C, and PL/SQL are trademarks of the Oracle Corporation. AMD and AMD Opteron are a trade mark of Advanced Micro Devices, Inc. HyperTransport is a licensed trademark of the HyperTransport Technology Consortium. TPC Benchmark, TPC-H, QppH, QthH and QphH are trademarks of the Transaction Processing Performance Council.

All other brand or product names mentioned herein must be considered trademarks or registered trademarks of their respective owners.

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark™ H test conducted on the HP BladeSystem ProLiant BL25p Cluster 8Pusing Oracle Database 10g Enterprise Edition with Real Application Cluster and Partitioning, in conformance with the requirements of the TPC Benchmark™ H Standard Specification, Revision 2.3.0. The operating system used for the benchmark was Red Hat Enterprise Linux 4 ES.

The benchmark results are summarized in the following table.

Hardware	Software	Total System Cost	QppH @ 300GB	QthH @ 300GB	QphH @ 300GB	$ / QphH @ 300GB
HP BladeSystem ProLiant BL25p Cluster 8P	Oracle Database 10g Enterprise Edition R2 with Real Application Cluster and Partitioning, and Red Hat Enterprise Linux 4 ES	$454,283	16,981.1	10,392.2	13284.2	$34.20

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

Standard and Executive Summary Statements

Executive Summary and Numerical Quantities Summary of the benchmark results for the HP BladeSystem ProLiant BL25p Cluster 8Pcan be found in the following pages.

Auditor

The benchmark configuration, environment and methodology were audited by Lorna Livingtree of Performance Metrics Inc. to verify compliance with the relevant TPC specifications.

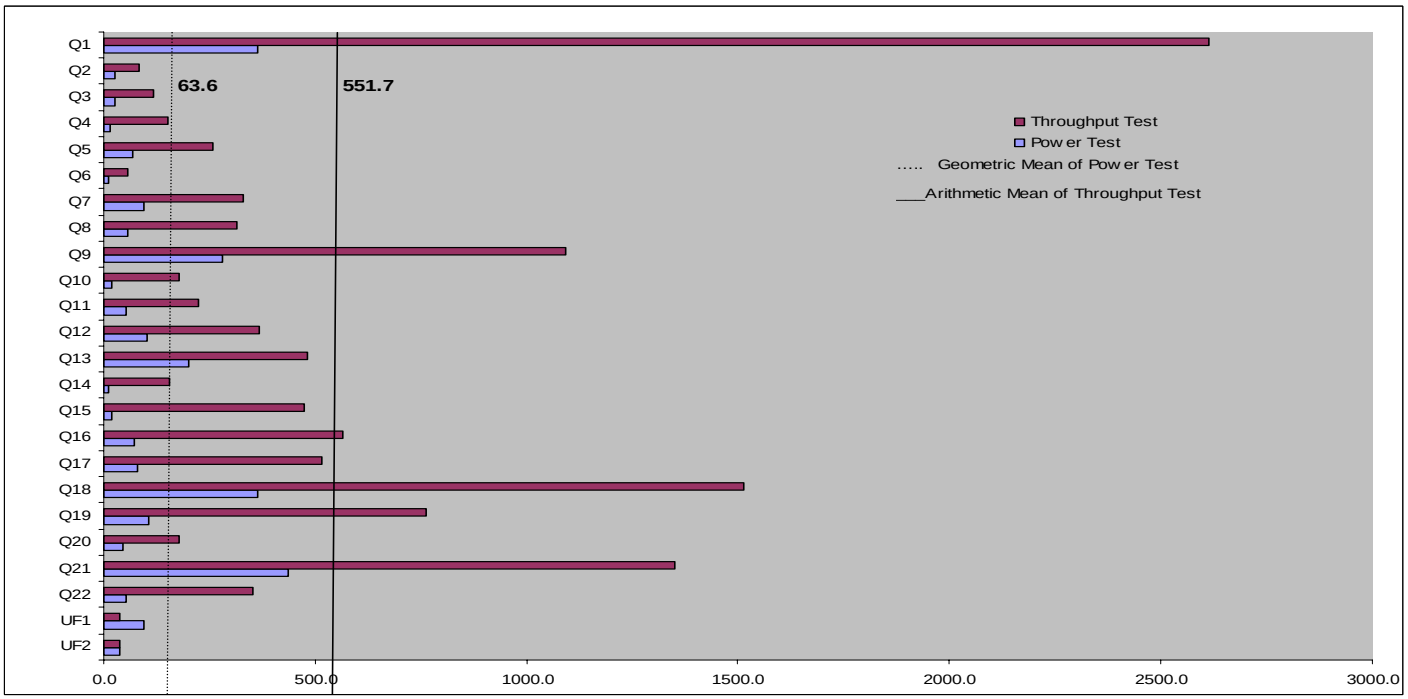
HP BladeSystem ProLiant BL25p Cluster 8P

TPC-H Rev. 2.3.0

Report Date: **September 16, 2005**

Total System Cost	Composite Query per Hour Metric	Price / Performance
$454,283	**13284.2** QphH@300GB	**$34.20 USD** $ / QphH@300GB

Database Size	Database Manager	Operating System	Other Software	Availability Date
300GB	**Oracle Database 10g Release 2, Enterprise Edition with Oracle Real Application Clusters and Partitioning**	**Red Hat Enterprise Linux 4 ES**		**October 31, 2005*** ***Hardware Available Now**



Database Load Time = 2:45:26	Load Included Backup: Y	Total Data Storage / Database Size = 15.36
RAID (Base tables only): N	RAID (Base tables and auxiliary data structures): N	RAID (All): N

8 ProLiant BL25p :

Processors (per blade) :	1 x AMD Opteron 252 2.6Ghz/1GHz Hyper Transport , 1MB processor
Cores (per blade) :	1
Threads (per blade) :	1
Memory (per blade) :	8 GB
OS Disk Drives (per blade) :	2 x 36GB 15krpm HDD Ultra320
Network (per blade) :	2 x on-board GigE (one of them as cluster interconnect)
Host Bus Adapter (per blade) :	2 x on-board
Storage Area Network :	2 x hp StorageWorks SAN Switch 2/16 16 x hp StorageWorks MSA1000 128 x 36GB 15krpm HDD Ultra320
Total Storage :	4608GB

HP BladeSystem ProLiant BL25p Cluster 8P

TPC-H Rev. 2.3.0

Report Date: 16-Sep-05

Description	Part Number	Brand	Pricing	Unit Price	Qty	Extended Price	3 yr. Maint. Price
Server Hardware							
HP BL25p O2.6GHz 1GB 1P Blade	374798-B21		1	2,699	8	21,592	
HP 4GB Reg PC3200 2x2GB Memory	379300-B21		1	2,049	16	32,784	
HP 5642 Unassembled Rack	358254-B21		1	689	2	1,378	
HP BL25/45p Fiber Channel Adapter	381881-B21		1	599	8	4,792	
HP T2200 XR High Voltage US UPS	204451-002		1	749	2	1,498	
HP BladeSystem p-Class Server Blade Enclosure	243564-B22		1	1,499	1	1,499	
HP ProLiant BL p-Class C-GbE2 Interconnect Kit	283192-B21		1	4,399	1	4,399	
HP BLp 1U Pwr Encl w/6 Pwr Spply Kit	378284-B21		1	2,299	1	2,299	
36GB 15Krpm U320 UNI HDD	286776-B22		1	299	16	4,784	
Blade Carepaq 3YR 4hr 24x7	UA205E		1	424	8		3,392
HP CAT5 KVM USB 1 Pack Interface Adapter	336047-B21		1	99	1	99	
HP s7540 17in. CRT Monitor	PF997AA#ABA		1	149	1	149	
USB Optical Mouse cbt/slvr USB 2-Button Optical Scroll Mouse	PT951AV		1	5	1	5	
Standard Keyboard USB	DX752AV#ABA		1	12	1	12	
					Subtotal	**75,290**	**3,392**
Storage							
hp Storageworks Modular SAN Array 1000	201723-B22		1	6,995	16	111,920	
hp StorageWorks Modular SAN Array 1000 Support 3YR 24x7 4HR	402164-002		1	3,538	16		56,608
HP StorageWorks SAN switch 2/16V	AA978A		1	9,990	2	19,980	
HP CP 3Y 4H 24x7 HW Edge Switch 2/16	340466-002		1	4,141	2	8,282	
5m LC to LC Cable Kit	221692-B22		1	82	32	2,624	
5m LC to LC Cable Kit (10% spares)	221692-B22		1	82	4	328	
36GB, 15krpm HDD Ultra320 HP	286776-B22		1	299	128	38,272	
36GB, 15krpm HDD Ultra320 HP (10% spares)	286776-B22		1	299	13	3,887	
2Gb SFF-SW Trncvr Kit ALL 2GB Small Form Pluggable Adapter Kit	221470-B21		1	199	32	6,368	
2Gb SFF-SW Trncvr Kit ALL 2GB Small Form Pluggable Adapter Kit (10% spares)	221470-B21		1	199	4	796	
					Subtotal	**192,457**	**56,608**
Software							
Oracle Database 10g Enterprise Edition Relase 2, Named User Plus for 3 Years	run-time	Oracle	2	10,000	8	80,000	
Oracle Real Application Clusters, Named User Plus for 3 Years	run-time	Oracle	2	5,000	8	40,000	
Partitioning, Named User Plus for 3 Years	run-time	Oracle	2	2,500	8	20,000	
Oracle Database Server Support Package for 3 Years	run-time	Oracle	2	16,000	3		48,000
Red Hat Enterprise Linux 4 ES	run-time	RedHat	3	799	8	6,392	
2 Addi. Yrs Subs. to Red Hat Enterprise Linux 4 ES	run-time	RedHat	3	799	16		12,784
					Subtotal	**146,392**	**60,784**
*hp Large Purchase discount			1			(42,840)	(9,600)
Oracle Mandatory E-Business Discount			2			(28,200)	
					Total	**343,099**	**111,184**
					3-Year Cost of Ownership:		**$454,283**
					QphH Rating:		**13284.2**
					$ / QphH@300GB:		**$34.20**

Price Key: 1-HP at 16% discount, 2-Oracle at 15% discount, 3-Red Hat, Oracle pricing contact: Mary Beth Pierantoni, mary.beth.pierantoni@oracle.com, 916-315-5081. *All discounts are based on US list prices and for similar quantities and configurations.

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

Numerical Quantities

Measurement Results:

Database Scale Factor	= 300
Total Data Storage / Database Size	= 15.36
Start of Database Load	= 9/7/2005 12:50:49
End of Database Load	= 9/7/2005 15:08:45
Database Load Time	= 02:17:56
Database Load Time + Backup Time	= 02:45:26

Query Streams for Throughput Test	= 6
TPC-H Power	= 16981.1
TPC-H Throughput	= 10392.2
TPC-H Composite Query-per-Hour Metric (QphH@300GB)	= 13284.2
Total System Price Over 3 Years	= $454,283
TPC-H Price/ Performance Metric ($/QphH@300GB)	= $34.20

Measurement Intervals:

Measurement Interval in Throughput Test (Ts)	= 13718.0 seconds

Duration of Stream Execution:

	Seed	Query Start Time Query End Time	RF1 Start Time RF1 End Time	RF2 Start Time RF2 End Time	Duration
Stream 0	907150845	09/07/2005 17:08:56	09/07/2005 17:07:21	09/07/2005 17:50:46	0:44:01
		09/07/2005 17:50:46	09/07/2005 17:08:56	09/07/2005 17:51:22	
Stream 1	907150846	09/07/2005 17:51:23	09/07/2005 21:32:06	09/07/2005 21:32:45	3:41:57
		09/07/2005 21:06:37	09/07/2005 21:32:45	09/07/2005 21:33:20	
Stream 2	907150847	09/07/2005 17:51:23	09/07/2005 21:33:20	09/07/2005 21:33:53	3:43:04
		09/07/2005 21:05:11	09/07/2005 21:33:53	09/07/2005 21:34:27	
Stream 3	907150848	09/07/2005 17:51:24	09/07/2005 21:34:27	09/07/2005 21:35:05	3:44:20
		09/07/2005 21:32:06	09/07/2005 21:35:05	09/07/2005 21:35:44	
Stream 4	907150849	09/07/2005 17:51:24	09/07/2005 21:35:44	09/07/2005 21:36:23	3:45:42
		09/07/2005 21:09:17	09/07/2005 21:36:23	09/07/2005 21:37:06	
Stream 5	907150850	09/07/2005 17:51:24	09/07/2005 21:37:06	09/07/2005 21:37:47	3:47:07
		09/07/2005 21:03:44	09/07/2005 21:37:47	09/07/2005 21:38:31	
Stream 6	907150851	09/07/2005 17:51:25	09/07/2005 21:38:31	09/07/2005 21:39:19	3:48:36
		09/07/2005 21:25:11	09/07/2005 21:37:47	09/07/2005 21:40:01	

HP BladeSystem ProLiant BL25p Cluster 8P

TPC-H Rev. 2.3.0

Report Date:
September 16, 2005

TPC-H Timing Intervals (in seconds)

Query	Stream 0	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Min Qi	Max Qi	Avg Qi
Q1	363.6	2616.7	2311.7	2741.1	2457.4	2708.8	2836.1	2311.7	2836.1	2612.0
Q2	25.2	97.4	96.2	50.4	61.6	105.4	86.4	50.4	105.4	82.9
Q3	28.0	204.5	90.9	110.6	51.9	136.5	188.2	51.9	204.5	130.4
Q4	16.4	71.9	215.3	264.7	210.3	179.9	132.9	71.9	264.7	179.2
Q5	69.8	229.4	272.8	81.2	474.5	149.3	156.1	81.2	474.5	227.2
Q6	12.6	87.1	42.6	364.1	45.8	54.9	34.0	34.0	364.1	104.8
Q7	94.3	696.8	219.9	308.3	302.7	200.4	184.9	184.9	696.8	318.8
Q8	57.4	173.5	329.4	1533.5	449.4	311.7	322.4	173.5	1533.5	520.0
Q9	282.4	425.3	968.8	208.9	1442.4	1255.7	926.1	208.9	1442.4	871.2
Q10	20.7	207.3	109.1	320.2	224.5	59.2	268.7	59.2	320.2	198.1
Q11	53.7	140.8	179.9	98.5	213.2	125.4	363.4	98.5	363.4	186.8
Q12	100.8	413.4	418.9	193.6	273.2	529.3	484.5	193.6	529.3	385.5
Q13	200.5	717.8	413.9	96.0	429.9	480.5	650.9	96.0	717.8	464.8
Q14	10.3	85.0	326.9	600.6	194.8	66.0	169.8	66.0	600.6	240.5
Q15	18.4	156.4	501.8	71.4	281.7	718.7	581.6	71.4	718.7	385.2
Q16	70.8	586.1	969.5	1209.7	660.9	587.2	515.2	515.2	1209.7	754.8
Q17	80.1	383.3	590.9	3409.0	282.7	155.5	467.4	155.5	3409.0	881.5
Q18	363.3	938.5	1170.4	479.1	1007.0	1186.7	1372.2	479.1	1372.2	1025.7
Q19	107.0	1343.2	1133.7	103.5	850.8	487.2	279.1	103.5	1343.2	699.6
Q20	43.7	263.6	103.0	589.6	92.0	89.4	426.9	89.4	589.6	260.7
Q21	436.7	1588.0	945.3	379.2	1527.2	1587.4	1857.6	379.2	1857.6	1314.1
Q22	54.5	288.0	216.4	29.2	338.9	364.9	522.5	29.2	522.5	293.3
UF1	93.8	38.8	32.4	38.6	38.6	41.6	47.9	32.4	47.9	39.7
UF2	36.5	36.0	33.9	39.5	42.6	43.7	42.1	33.9	43.7	39.7

Table Of Contents

1.0 General Items

1.1 Test Sponsor

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

Hewlett Packard Company sponsored this benchmark. The benchmark was developed and engineered by Hewlett Packard Company and Oracle Corporation. Testing took place at HP Database Performance Engineering Laboratory in Houston, Texas.

1.2 Parameter Settings

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

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

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

Appendix A contains Database and Operating system parameter settings.

1.3 Configuration Items

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

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

The HP BladeSystem ProLiant BL25p Cluster 8P is depicted in Figure 1.1 consists of:

Processors (per blade) :1 x AMD Opteron 252 2.6Ghz/1GHz Hyper Transport , 1MB processor

Cores (per blade) :	1
Threads (per blade) :	1
Memory (per blade) :	8 GB
OS Disk Drives (per blade) :	2 x 36GB 15krpm HDD Ultra320
Network (per blade) :	2 x on-board GigE (one of them as cluster interconnect)
Host Bus Adapter (per blade) :	2 x on-board
Storage Area Network :	2 x hp StorageWorks SAN Switch 2/16 16 x hp StorageWorks MSA1000 128 x 36GB 15krpm HDD Ultra320
Total Storage :	4608GB

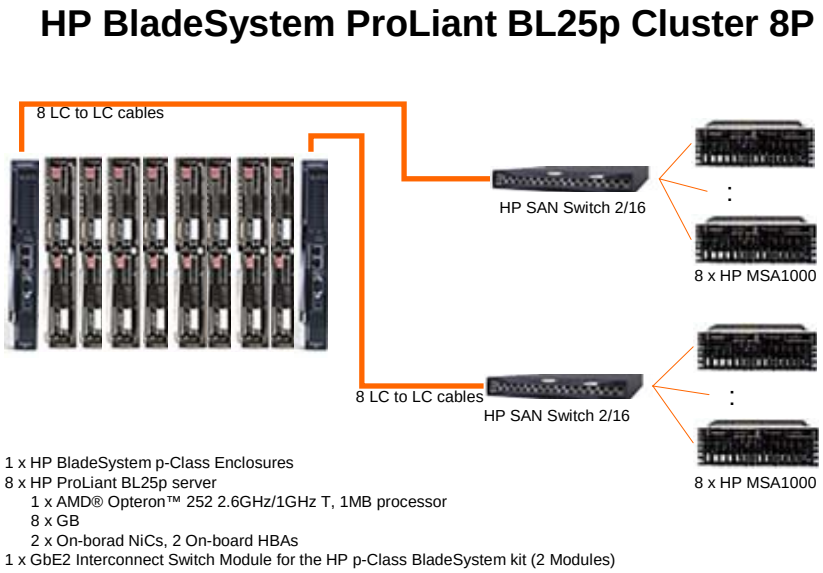
The storage area network (SAN) consists of 2 hp SAN Switch 2/16s and 16 hp StorageWorks MSA1000s. The hp ProLiant BL25p Server Blade has two on-board HBAs, connected to one of two hp SAN Switch 2/16s, each hp SAN Switch 2/16s has eight hp StorageWorks MSA1000s connected to it.

The hp ProLiant BL25p Server Blade has two on-board NICs; one used as Oracle 10g cluster interconnect and other as cluster manager communication and user access.

Each hp StorageWorks MSA1000 has two RAID0 volumes of four 36GB 15krpm HDD Ultra320s. Each RAID0 volume is partitioned using Linux. The tables and indexes reside on an Oracle 10g ASM disk group consisted of one Linux partition from each RAID0 volume; the temp tablespace resides on another Oracle 10g ASM disk group which also consisted of one Linux partition from each RAID0 volume. All tables except NATION and REGION have 32 horizontal partitions. One Linux partition from each RAID0 volume is used for redo log files. There are 16 redo log file groups; each redo file group has two members residing on two separate MSA1000s to guarantee no single point of controller/cache failure. The MSA1000 array accelerator cache is set to 100% write.

A detailed description of distribution of database files can be found in Table 5.2.

Figure 1.1: Benchmarked and Priced Configuration



2.0 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. (8.1.2.1)

Appendix B contains the database build scripts.

2.2 Physical Organization of Database

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

Please refer Appendix B for column reordering of tables.

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 as described in Appendix B.

2.4 Replication

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

The database was not replicated.

3.0 Clause 2: Queries and Refresh Functions Related Items

3.1 Query Language

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

SQL was the query language used to implement all queries.

3.2 Random Number Generation

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

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

3.3 Substitution Parameters Generation

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

The supplied QGEN version 1.3.0 was used to generate the substitution parameters.

3.4 Query Text and Output Data from Database

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

Appendix D contains the query text and output.

3.5 Query Substitution Parameters and Seeds Used

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

Appendix E contains the query substitution parameters and seed used.

3.6 Isolation Level

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

The queries and transactions were run with the isolation level set to “Level 3” (repeatable read).

3.7 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 implementation-specific layer/driver code included in Appendix F.

4.0 Clause 3: Database System Properties

4.1 Atomicity Requirements

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

Appendix C contains the source code for the ACID transactions.

4.1.1 Atomicity of the Completed Transactions

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

The following steps were performed to verify the Atomicity of the completed ACID transactions:

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.

4.1.2 Atomicity of Aborted Transactions

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

The following steps were performed to verify the Atomicity of the aborted ACID transactions:

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.

4.2 Consistency Requirements

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

A consistent state for the TPC-H database is defined to exist when:

*O_TOTALPRICE = SUM(L_EXTENDEDPRICE – L_DISCOUNT) * (1 + L_TAX)*
For each ORDER and LINEITEM defined by (O_ORDERKEY = L_ORDERKEY)

The following queries were executed before and after a measurement to show that the database was always in a consistent state both initially and after a measurement.

SELECT DECIMAL (SUM (DECIMAL (INTEGER (INTEGER (DECIMAL (INTEGER (100 * DECIMAL (L_EXTENDEDPRICE, 20, 3)), 20, 3) * (1 – L_DISCOUNT)) * (1 + L_TAX)), 20, 3) / 100.0) 20, 3) FROM TPCD.LINEITEM WHERE L_ORDERKEY = okey

SELECT DECIMAL(SUM(O_TOTALPRICE, 20, 3)) from TPCH.ORDERS WHERE O_ORDERKEY = okey

4.2.1 Consistency Tests

Verify that ORDER and LINEITEM tables are initially consistent as defined in Clause 3.3.2.1, based upon a random sample of at least 10 distinct values of O_ORDERKEY.

The following steps were performed to verify the Consistency of ACID transactions:

1. The consistency of the ORDERS and LINEITEM tables was verified based on a sample of order keys.
2. 100 ACID Transactions were submitted from each of 8 execution streams.
3. The consistency of the ORDERS and LINEITEM tables was re-verified.

4.3 Isolation Requirements

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

4.3.1 Isolation Test 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.

The following steps were performed to satisfy the test of isolation for a read-only and a read-write committed transaction:

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 any 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.

4.3.2 Isolation Test 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.

The following steps were performed to satisfy the test of isolation for a read-only and a rolled back read-write transaction:

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.

4.3.3 Isolation Test 3 - Write-Write Conflict with Commit

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

The following steps were performed to verify isolation of two update transactions:

1. An ACID Transaction, T1, was started for a randomly selected O_KEY, L_KEY, and DELTA. The ACID transaction 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))

4.3.4 Isolation Test 4 - Write-Write Conflict with Rollback

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

The following steps were performed to verify isolation of two update transactions after the first one is rolled back:

1. An ACID Transaction, T1, was started for a randomly selected O_KEY, L_KEY, and DELTA. The ACID transaction 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.

4.3.5 Isolation Test 5 – Concurrent Read and Write Transactions on Different Tables

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

The following steps were performed to 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 COMMIT.
2. Another ACID transaction, T2 was started using random values for PS_PARTKEY and PS_SUPPKEY, all columns of the PARTSUPP table for which PS_PARTKEY and PS_SUPPKEy are equal are returned.
3. ACID Transaction T2 completed.
4. T1 was allowed to COMMIT.
5. It was verified that the appropriate rows in the ORDER, LINEITEM, and HISTORY tables have been changed.

4.3.6 Isolation Test 6 – Update Transactions during Continuous Read-Only Query Stream

Demonstrate 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.

The following steps were performed to demonstrate that the continuous submission of arbitrary (read-only) queries against one or more tables of the database:

1. A Transaction, T1, was started which executed Q21 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 Q21.

4.4 Durability Requirements

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

4.4.1 Permanent Unrecoverable Failure of Any 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.

Qualification database was brought up on two nodes. Started test transactions. During the test one of the redo log files (each redo log file belongs to a group which has two members residing on two separate MSA1000s) was corrupted by writing garbage information onto it. The database reported file corruption and the corresponding database instance terminated. Database Instance was shutdown on the second node. Started Oracle instance, which automatically recovered the database from the clean redo log file. Consistency conditions were verified.

One of the data files was backup up. Qualification database was brought up on two nodes. Started test transactions. During the test, the backed up data file was corrupted by writing garbage information onto it. The database reported file corruption and instances were terminated. The data file was restored from the backup. Started Oracle instance, which automatically recovered the database. Consistency conditions were verified.

4.4.2 System Crash

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

The system crash and memory failure tests were combined. One of the data files was backup up. Qualification database was brought up on two nodes. Started test transactions. During the test power to the HP BladeSystem enclosure was turned off. The power was restored. Started Oracle instance, which automatically recovered the database. The durability success file and the HISTORY table were compared and the counts matched.

4.4.3 Memory Failure

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

The system crash and memory failure tests were combined as explained in section 4.4.2.

5.0 Clause 4: Scaling and Database Population

5.1 Initial Cardinality of Tables

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

Table 5.1 lists the TPC Benchmark H defined tables and the row count for each table as they existed upon completion of the build.

Table Name	Row Count
Region	5
Nation	25
Supplier	3,000,000
Customer	45,000,000
Part	60,000,000
Partsupp	240,000,000
Orders	450,000,000
Lineitem	1,799,989,091

Table 5. 1: Initial Number of Rows

5.2 Distribution of Tables and Logs Across Media

The distribution of tables and logs across all media must be explicitly described for the tested and priced systems.

The HP BladeSystem ProLiant BL25p Cluster 8P is depicted in Figure 1.1 consists of:

Processors (per blade) :1 x AMD Opteron 252 2.6Ghz/1GHz Hyper Transport , 1MB processor

Cores (per blade) :	1
Threads (per blade) :	1
Memory (per blade) :	8 GB
OS Disk Drives (per blade) :	2 x 36GB 15krpm HDD Ultra320
Network (per blade) :	2 x on-board GigE (one of them as cluster interconnect)
Host Bus Adapter (per blade) :	2 x on-board
Storage Area Network :	2 x hp StorageWorks SAN Switch 2/16 16 x hp StorageWorks MSA1000 128 x 36GB 15krpm HDD Ultra320
Total Storage :	4608GB

The storage area network (SAN) consists of 2 hp SAN Switch 2/16s and 16 hp StorageWorks MSA1000s. The hp ProLiant BL25p Server Blade has two on-board HBAs, connected to one of two hp SAN Switch 2/16s, each hp SAN Switch 2/16s has eight hp StorageWorks MSA1000s connected to it.

The hp ProLiant BL25p Server Blade has two on-board NICs; one used as Oracle 10g cluster interconnect and other as cluster manager communication and user access.

Each hp StorageWorks MSA1000 has two RAID0 volumes of four 36GB 15krpm HDD Ultra320s. Each RAID0 volume is partitioned using Linux. The tables and indexes reside on an Oracle 10g ASM disk group consisted of one Linux partition from each RAID0 volume; the temp tablespace resides on another Oracle 10g ASM disk group which also consisted of one Linux partition from each RAID0 volume. All tables except NATION and REGION have 32 horizontal partitions. One Linux partition from each RAID0 volume is used for redo log files. There are 16 redo log file groups; each redo file group has two members residing on two separate MSA1000s to guarantee no single point of controller/cache failure. The MSA1000 array accelerator cache is set to 100% write.

A detailed description of distribution of database files can be found in Table 5.2.

SAN Switch / MSA 1000	Array	Linux Partition	Type	Description	
SAN Switch 1 - MSA [1..8] SAN Switch 2 – MSA [9..16]	1	1	raw	ASM Disk Group DATA (tables and indexes) (1/32)	20480MB
		2	raw	ASM Disk Group TEMP (1/32)	10240MB
		3		Unused	
		5	raw	*misc (1 of 8)	256MB
		6	raw	redo log (1/32)	4096MB
		7	ext3	flat files (1/32)	12802MB
		8	ext3	Backup (1/32)	47693MB
	2	1	raw	ASM Disk Group DATA (tables and indexes) (1/32)	20480MB
		2	raw	ASM Disk Group TEMP (1/32)	10240MB
		3		Unused	
		5	raw	undo (1/8)	8192MB
		6	raw	redo log (1/32)	4096MB
		7	ext3	flat files (1/32)	12802MB
		8	ext3	Backup (1/32)	47693MB
*misc - ocr, quorum, control1, control2, sys, sysaux, sp_0, default					

Table 5.2: SAN configuration and Database Layout

5.3 Mapping of Database Partitions/Replications

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

Horizontal partitioning was used for all tables except NATION and REGION. Sections 5.2 describe the distribution of tables and redo log files. The database was not replicated.

5.4 Implementation of RAID

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

RAID 0 was used for the entire database and redo log files. Each redo log group members were placed on two separate MSA1000s.

5.5 DBGEN Modifications

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

The supplied DBGEN version 1.3.0 was used to generate the database population for this benchmark without any modification.

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 2 hours 45 minutes 26 seconds, includes time to backup database files.

5.7 Data Storage Ratio

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

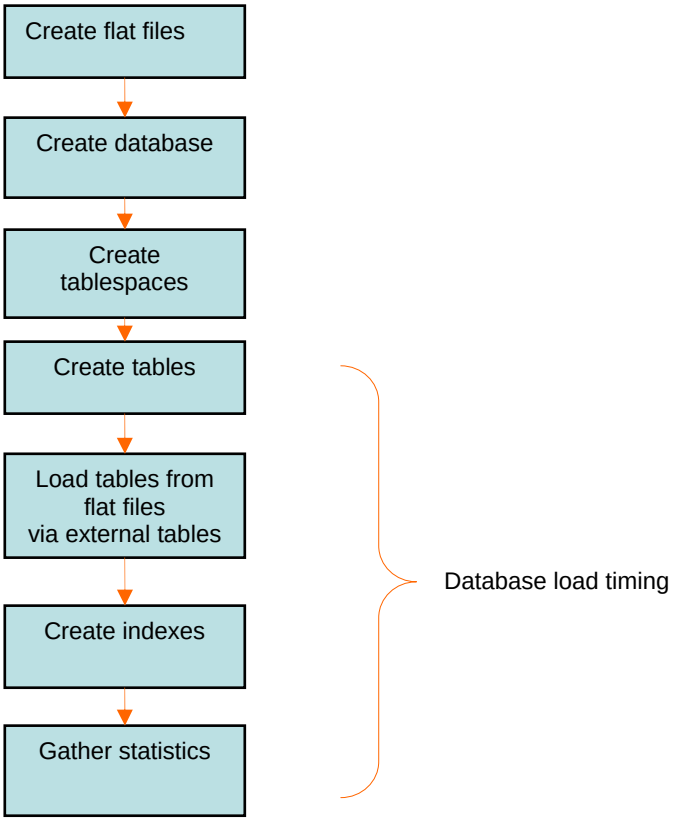
Disk Type	Number of Disks	Total Disk Space	Data Storage Ratio
36GB15krpm HDD Ultra320	128	4608GB	15.36

5.8 Database Load Mechanism Details and Illustration

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

Flat files for each of the tables were created using DBGEN resided on the SAN.

Figure 5.8: Block Diagram of Database Load Process



5.9 Qualification Database Configuration

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

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

6.0 Clause 5: Performance Metrics and Execution Rules Related Items

6.1 Steps in the Power Test

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

The following steps were used to implement the power test:

1. The system was rebooted
2. RF1 Refresh Transaction
3. Stream 00 Execution
4. RF2 Refresh Transaction.

6.2 Timing Intervals for Each Query and Refresh Function

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

Numerical Quantities Summary section of the executive summary, which can be found in the beginning of this document, contains the timing intervals for queries and refresh functions.

6.3 Number of Streams for The Throughput Test

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

Six streams were used for the Throughput Test.

6.4 Start and End Date/Times for Each Query Stream

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

Numerical Quantities Summary section of the executive summary, which can be found in the beginning of this document, contains the start and stop times for the query execution streams.

6.5 Total Elapsed Time for the Measurement Interval

The total elapsed time of the measurement interval (see Clause 5.3.5) must be reported for the throughput test.

Numerical Quantities Summary section of the executive summary, which can be found in the beginning of this document, contains the elapsed time for the measurement interval.

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

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

Numerical Quantities Summary section of the executive summary, which can be found in the beginning of this document, contains the start and finish time for the refresh functions.

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

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

Numerical Quantities Summary section of the executive summary, which can be found in the beginning of this document, contains the timing intervals for queries and refresh functions.

6.8 Performance Metrics

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

Numerical Quantities Summary section of the executive summary, which can be found in the beginning of this document, contains the performance metrics, related numerical quantities and the price performance metric.

6.9 The Performance Metric and Numerical Quantities from Both Runs

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

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

Run	QppH@300GB	QthH@300GB	QphH@300GB
Run 1	16,981.1	10,392.2	13,284.2
Run 2	17142.9	10,490.8	13,410.5

6.11 System Activity Between Tests

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

No activites performed between Run 1 and Run 2.

7.0 Clause 6: SUT and Driver Implementation Related Items

7.1 Driver

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

A single script performs all stream executions. QGEN is used to produce query text. For each power-test run:
The SQL for RF1 is submitted to the database
Then the queries as generated by QGEN are submitted in the order defined by Clause 5.3.5.4
The SQL for RF2 is submitted to the database.

7.2 Implementation Specific Layer (ISL)

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

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

7.3 Profile-Directed Optimization

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

Profile-directed optimization was used in this benchmark.

8.0 Clause 7: Pricing Related Items

8.1 Hardware and Software Used

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

A detailed list of hardware and software used in the priced system is included in the pricing sheet in the executive summary. All prices are currently effective. Third-party price quotations are included in Appendix G.

8.2 Total 3 Year Price

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

A detailed pricing sheet of all the hardware and software used in this configuration and the 3-year maintenance costs, demonstrating the computation of the total 3-year price of the configuration, is included in the executive summary. This purchase qualifies for 16% large purchase discount from Hewlett Packard Company. Oracle Database software qualify for an Oracle mandatory E-Business discount.

8.3 Availability Date

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

Availability date is October 31, 2005. Hardware is available at the time of publication.

8.4 Country-Specific Pricing

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

The configuration is priced for the United States of America.

9.0 Clause 9: Related Items

9.1 Auditors' Report

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

Lorna Livingtree of Performance Metrics Inc audited this implementation of the TPC Benchmark H.

Performance Metrics Inc.
137 Yankton St., Suite 101
Folsom, CA 95630
USA

Email: lorna@perfmetrics.com

TPC Benchmark H Full Disclosure Report and other information can be downloaded from Transaction Processing Performance Council web site at www.tpc.org.

PERFORMANCE METRICS INC.
TPC Certified Auditors

September 13, 2005

Mr. Raghunath Othayoth
Hewlett-Packard Company
20555 SH 249
Houston, TX 77077

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

Platform: ProLiant BL25p 8 node cluster
Database Manager: Oracle Database 10g Enterprise Edition R2
Operating System: Red Hat Enterprise Linux ES

CPU's	Memory	Total Disks	Qpph@ 1000GB	QthH@1000GB	QphH@1000GB
8 AMD Opterons @ 2.6 Ghz	8 GB each node	16 @ 36 GB (OS) 128 @ 36GB	**16,981.1**	**10,392.2**	**13,284.2**

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

- The database tables were defined with the proper columns, layout and sizes.
- The tested database was correctly scaled and populated for 300GB using DBGEN. The version of DBGEN was 1.3.0.
- The qualification database layout was identical to the tested database except for the number and size of the files and nodes.
- The query text was verified to use only compliant variants and minor modifications.
- The executable query text was generated by QGEN and submitted through Oracle's standard interactive interface. The version of QGEN was 1.3.0.
- The validation of the query text against the qualification database produced compliant results.
- The refresh functions were properly implemented and executed the correct number of inserts and deletes.

- The load timing was properly measured and reported.
- The execution times were correctly measured and reported.
- The performance metrics were correctly computed and reported.
- The repeatability of the measurement was verified.
- The ACID properties were successfully demonstrated and verified.
- Sufficient mirrored log space was present on the tested system.
- The system pricing was checked for major components and maintenance.
- The executive summary pages of the FDR were verified for accuracy.

Auditor's Notes:
None.

Sincerely,

Lorna Livingtree

Lorna Livingtree
Auditor

Appendix A: Parameter Settings

```
-----------------------------------------
init+ASM1.ora
-----------------------------------------
###############################################################################
# Copyright (c) 1991, 2001, 2002 by Oracle Corporation
###############################################################################

###########################################
# Cluster Database
###########################################
cluster_database=true

###########################################
# Diagnostics and Statistics
###########################################
background_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/bdump
core_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/cdump
user_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/udump

###########################################
# Miscellaneous
###########################################
instance_type=asm

###########################################
# Pools
###########################################
shared_pool_size=120M
large_pool_size=12M

###########################################
# Security and Auditing
###########################################
remote_login_passwordfile=exclusive

asm_diskgroups='DATA','TEMP','DATA1g'
+ASM8.instance_number=8
+ASM7.instance_number=7
+ASM6.instance_number=6
+ASM5.instance_number=5
+ASM4.instance_number=4
+ASM3.instance_number=3
+ASM2.instance_number=2
+ASM1.instance_number=1
instance_number = 1
processes = 500
ASM_DISKSTRING='/home/oracle/dev/raw/*'


-----------------------------------------
init+ASM2.ora
-----------------------------------------
###############################################################################
# Copyright (c) 1991, 2001, 2002 by Oracle Corporation
###############################################################################

###########################################
# Cluster Database
###########################################
cluster_database=true

###########################################
# Diagnostics and Statistics
###########################################
background_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/bdump
core_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/cdump
user_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/udump

###########################################
# Miscellaneous
###########################################
instance_type=asm

###########################################
# Pools
###########################################
shared_pool_size=120M
large_pool_size=12M

###########################################
# Security and Auditing
###########################################
remote_login_passwordfile=exclusive

asm_diskgroups='DATA','TEMP','DATA1g'
+ASM8.instance_number=8
+ASM7.instance_number=7
+ASM6.instance_number=6
+ASM5.instance_number=5
+ASM4.instance_number=4
+ASM3.instance_number=3
+ASM2.instance_number=2
+ASM1.instance_number=1
instance_number = 2
processes = 500
ASM_DISKSTRING='/home/oracle/dev/raw/*'


-----------------------------------------
init+ASM3.ora
-----------------------------------------
###############################################################################
# Copyright (c) 1991, 2001, 2002 by Oracle Corporation
###############################################################################

###########################################
# Cluster Database
###########################################
cluster_database=true

###########################################
# Diagnostics and Statistics
###########################################
background_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/bdump
core_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/cdump
user_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/udump

###########################################
# Miscellaneous
###########################################
instance_type=asm

###########################################
# Pools
###########################################
shared_pool_size=120M
large_pool_size=12M

###########################################
# Security and Auditing
###########################################
remote_login_passwordfile=exclusive

asm_diskgroups='DATA','TEMP'
+ASM8.instance_number=8
+ASM7.instance_number=7
+ASM6.instance_number=6
+ASM5.instance_number=5
+ASM4.instance_number=4
+ASM3.instance_number=3
+ASM2.instance_number=2
+ASM1.instance_number=1
instance_number = 3
processes = 500
ASM_DISKSTRING='/home/oracle/dev/raw/*'


-----------------------------------------
init+ASM4.ora
-----------------------------------------
###############################################################################
# Copyright (c) 1991, 2001, 2002 by Oracle Corporation
###############################################################################

###########################################
# Cluster Database
###########################################
cluster_database=true

###########################################
# Diagnostics and Statistics
###########################################
background_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/bdump
core_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/cdump
user_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/udump

###########################################
# Miscellaneous
###########################################
instance_type=asm

###########################################
# Pools
###########################################
```

```
shared_pool_size=120M
large_pool_size=12M

########################################
# Security and Auditing
########################################
remote_login_passwordfile=exclusive

asm_diskgroups='DATA','TEMP'
+ASM8.instance_number=8
+ASM7.instance_number=7
+ASM6.instance_number=6
+ASM5.instance_number=5
+ASM4.instance_number=4
+ASM3.instance_number=3
+ASM2.instance_number=2
+ASM1.instance_number=1
instance_number = 4
processes = 500
ASM_DISKSTRING='/home/oracle/dev/raw/*'


-----------------------------------------
init+ASM5.ora
-----------------------------------------
##############################################################################
# Copyright (c) 1991, 2001, 2002 by Oracle Corporation
##############################################################################

########################################
# Cluster Database
########################################
cluster_database=true

########################################
# Diagnostics and Statistics
########################################
background_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/bdump
core_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/cdump
user_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/udump

########################################
# Miscellaneous
########################################
instance_type=asm

########################################
# Pools
########################################
shared_pool_size=120M
large_pool_size=12M

########################################
# Security and Auditing
########################################
remote_login_passwordfile=exclusive

asm_diskgroups='DATA','TEMP'
+ASM8.instance_number=8
+ASM7.instance_number=7
+ASM6.instance_number=6
+ASM5.instance_number=5
+ASM4.instance_number=4
+ASM3.instance_number=3
+ASM2.instance_number=2
+ASM1.instance_number=1
instance_number = 5
processes = 500
ASM_DISKSTRING='/home/oracle/dev/raw/*'


-----------------------------------------
init+ASM6.ora
-----------------------------------------
##############################################################################
# Copyright (c) 1991, 2001, 2002 by Oracle Corporation
##############################################################################

########################################
# Cluster Database
########################################
cluster_database=true

########################################
# Diagnostics and Statistics
########################################
background_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/bdump
core_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/cdump
user_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/udump

########################################
# Miscellaneous
########################################
instance_type=asm

########################################
# Pools
########################################
shared_pool_size=120M
large_pool_size=12M

########################################
# Security and Auditing
########################################
remote_login_passwordfile=exclusive

asm_diskgroups='DATA','TEMP'
+ASM8.instance_number=8
+ASM7.instance_number=7
+ASM6.instance_number=6
+ASM5.instance_number=5
+ASM4.instance_number=4
+ASM3.instance_number=3
+ASM2.instance_number=2
+ASM1.instance_number=1
instance_number = 6
processes = 500
ASM_DISKSTRING='/home/oracle/dev/raw/*'


-----------------------------------------
init+ASM7.ora
-----------------------------------------
##############################################################################
# Copyright (c) 1991, 2001, 2002 by Oracle Corporation
##############################################################################

########################################
# Cluster Database
########################################
cluster_database=true

########################################
# Diagnostics and Statistics
########################################
background_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/bdump
core_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/cdump
user_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/udump

########################################
# Miscellaneous
########################################
instance_type=asm

########################################
# Pools
########################################
shared_pool_size=120M
large_pool_size=12M

########################################
# Security and Auditing
########################################
remote_login_passwordfile=exclusive

asm_diskgroups='DATA','TEMP'
+ASM8.instance_number=8
+ASM7.instance_number=7
+ASM6.instance_number=6
+ASM5.instance_number=5
+ASM4.instance_number=4
+ASM3.instance_number=3
+ASM2.instance_number=2
+ASM1.instance_number=1
instance_number = 7
processes = 500
ASM_DISKSTRING='/home/oracle/dev/raw/*'


-----------------------------------------
init+ASM8.ora
-----------------------------------------
##############################################################################
# Copyright (c) 1991, 2001, 2002 by Oracle Corporation
##############################################################################

########################################
# Cluster Database
########################################
cluster_database=true

########################################
# Diagnostics and Statistics
########################################
background_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/bdump
core_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/cdump
```

```
user_dump_dest=/home/oracle/10gR2_rel/admin/+ASM/udump

###########################################
# Miscellaneous
###########################################
instance_type=asm

###########################################
# Pools
###########################################
shared_pool_size=120M
large_pool_size=12M

###########################################
# Security and Auditing
###########################################
remote_login_passwordfile=exclusive

asm_diskgroups='DATA','TEMP'
+ASM8.instance_number=8
+ASM7.instance_number=7
+ASM6.instance_number=6
+ASM5.instance_number=5
+ASM4.instance_number=4
+ASM3.instance_number=3
+ASM2.instance_number=2
+ASM1.instance_number=1
instance_number = 8
processes = 500
ASM_DISKSTRING='/home/oracle/dev/raw/*'


---------------------------------------------
init_blh1.ora
---------------------------------------------
instance_number              = 1
thread                       = 1
undo_management              = auto
undo_tablespace              = ts_undo1
cluster_database             = true
cluster_interconnects        = 1.1.204.21
#cluster_interconnects        = 10.1.204.21
ifile                        = /home/oracle/10gR2_rel/db/dbs/init_build.ora
#ifile                       = /home/oracle/10gR2_rel/db/dbs/init_noawr.ora



---------------------------------------------
init_blh2.ora
---------------------------------------------
instance_number              = 2
thread                       = 2
undo_management              = auto
undo_tablespace              = ts_undo2
cluster_database             = true
cluster_interconnects        = 1.1.204.22
#cluster_interconnects        = 10.1.204.22
ifile                        = /home/oracle/10gR2_rel/db/dbs/init_build.ora
#ifile                       = /home/oracle/10gR2_rel/db/dbs/init_noawr.ora



---------------------------------------------
init_blh3.ora
---------------------------------------------
instance_number              = 3
thread                       = 3
undo_management              = auto
undo_tablespace              = ts_undo3
cluster_database             = true
cluster_interconnects        = 1.1.204.23
#cluster_interconnects        = 10.1.204.23
ifile                        = /home/oracle/10gR2_rel/db/dbs/init_build.ora
#ifile                       = /home/oracle/10gR2_rel/db/dbs/init_noawr.ora



---------------------------------------------
init_blh4.ora
---------------------------------------------
instance_number              = 4
thread                       = 4
undo_management              = auto
undo_tablespace              = ts_undo4
cluster_database             = true
cluster_interconnects        = 1.1.204.24
#cluster_interconnects        = 10.1.204.24
ifile                        = /home/oracle/10gR2_rel/db/dbs/init_build.ora
#ifile                       = /home/oracle/10gR2_rel/db/dbs/init_noawr.ora

---------------------------------------------
init_blh5.ora
---------------------------------------------
instance_number              = 5
thread                       = 5
undo_management              = auto
undo_tablespace              = ts_undo5
cluster_database             = true
cluster_interconnects        = 1.1.204.25
#cluster_interconnects        = 10.1.204.25
ifile                        = /home/oracle/10gR2_rel/db/dbs/init_build.ora
#ifile                       = /home/oracle/10gR2_rel/db/dbs/init_noawr.ora



---------------------------------------------
init_blh6.ora
---------------------------------------------
instance_number              = 6
thread                       = 6
undo_management              = auto
undo_tablespace              = ts_undo6
cluster_database             = true
cluster_interconnects        = 1.1.204.26
#cluster_interconnects        = 10.1.204.26
ifile                        = /home/oracle/10gR2_rel/db/dbs/init_build.ora
#ifile                       = /home/oracle/10gR2_rel/db/dbs/init_noawr.ora



---------------------------------------------
init_blh7.ora
---------------------------------------------
instance_number              = 7
thread                       = 7
undo_management              = auto
undo_tablespace              = ts_undo7
cluster_database             = true
cluster_interconnects        = 1.1.204.27
#cluster_interconnects        = 10.1.204.27
ifile                        = /home/oracle/10gR2_rel/db/dbs/init_build.ora
#ifile                       = /home/oracle/10gR2_rel/db/dbs/init_noawr.ora



---------------------------------------------
init_blh8.ora
---------------------------------------------
instance_number              = 8
thread                       = 8
undo_management              = auto
undo_tablespace              = ts_undo8
cluster_database             = true
cluster_interconnects        = 1.1.204.28
#cluster_interconnects        = 10.1.204.28
ifile                        = /home/oracle/10gR2_rel/db/dbs/init_build.ora
#ifile                       = /home/oracle/10gR2_rel/db/dbs/init_noawr.ora



---------------------------------------------
init_build.ora
---------------------------------------------
aq_tm_processes              = 0
audit_trail                  = false
compatible                   = 10.1.0.2
control_files                = (/home/oracle/dev/raw/control1,  /home/oracle/dev/raw/control2)
db_block_checksum            = false
db_block_size                = 16384
db_file_multiblock_read_count  = 64
db_files                     = 500
db_name                      = 10i
db_writer_processes          = 4
dml_locks                    = 5000
global_names                 = false
instance_name                = raca
log_buffer                   = 4194304
log_checkpoints_to_alert     = true
max_dump_file_size           = unlimited
nls_date_format              = YYYY-MM-DD
open_cursors                 = 600
optimizer_mode               = CHOOSE
optimizer_features_enable    = 10.2.0.1.1
parallel_adaptive_multi_user  = true
parallel_execution_message_size = 32768
parallel_max_servers         = 100
parallel_min_servers         = 64
pga_aggregate_target         = 5500m
processes                    = 1000
query_rewrite_integrity      = stale_tolerated
recovery_parallelism         = 8
replication_dependency_tracking = false
sga_target                   = 2400m
```

```
statistics_level                = typical
undo_management                 = auto
undo_retention                  = 400000
optimizer_index_cost_adj        = 450


---------------------------------------------
/boot/grub/grub.conf
---------------------------------------------
#          kernel /vmlinuz-version ro root=/dev/VolGroup01/LogVol00
#          initrd /initrd-version.img
#boot=/dev/cciss/c0d0
default=1
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux AS (2.6.9-11.EL)
        root (hd0,0)
        kernel /vmlinuz-2.6.9-11.EL ro root=/dev/VolGroup01/LogVol00 rhgb quiet maxcpus=1
elevator=deadline
        initrd /initrd-2.6.9-11.EL.img
title Red Hat Enterprise Linux AS (2.6.9-11.ELsmp)
        root (hd0,0)
        kernel /vmlinuz-2.6.9-11.ELsmp ro root=/dev/VolGroup01/LogVol00 rhgb quiet
maxcpus=1 elevator=noop
        initrd /initrd-2.6.9-11.ELsmp.img

---------------------------------------------
/etc/sysctl.conf
---------------------------------------------
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1

kernel.sem = 250        32000   100      128
net.core.rmem_max = 524288
net.core.rmem_default = 524288
net.core.wmem_max = 524288
net.core.wmem_default = 524288
fs.aio-max-nr = 2097152
kernel.shmmax = 3000000000

---------------------------------------------
/etc/rc.local

#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local

chown -R oracle /home/oracle/dev/*

echo 1300 > /proc/sys/vm/nr_hugepages
echo 500 > /proc/sys/vm/hugetlb_shm_group
```

Appendix B: Database Build Scripts

```
---------------------------------------------
build_dg_10gR2.sh
---------------------------------------------
export ORACLE_SID=+ASM1
sqlplus /NOLOG <<!
connect /as sysdba
drop diskgroup DATA including contents;
drop diskgroup TEMP including contents;
CREATE DISKGROUP DATA External REDUNDANCY  DISK
'/home/oracle/dev/raw/lo_1' SIZE 20480M ,
'/home/oracle/dev/raw/lo_2' SIZE 20480M ,
'/home/oracle/dev/raw/lo_3' SIZE 20480M ,
'/home/oracle/dev/raw/lo_4' SIZE 20480M ,
'/home/oracle/dev/raw/lo_5' SIZE 20480M ,
'/home/oracle/dev/raw/lo_6' SIZE 20480M ,
'/home/oracle/dev/raw/lo_7' SIZE 20480M ,
'/home/oracle/dev/raw/lo_8' SIZE 20480M ,
'/home/oracle/dev/raw/lo_9' SIZE 20480M ,
'/home/oracle/dev/raw/lo_10' SIZE 20480M ,
'/home/oracle/dev/raw/lo_11' SIZE 20480M ,
'/home/oracle/dev/raw/lo_12' SIZE 20480M ,
'/home/oracle/dev/raw/lo_13' SIZE 20480M ,
'/home/oracle/dev/raw/lo_14' SIZE 20480M ,
'/home/oracle/dev/raw/lo_15' SIZE 20480M ,
'/home/oracle/dev/raw/lo_16' SIZE 20480M ,
'/home/oracle/dev/raw/lo_17' SIZE 20480M ,
'/home/oracle/dev/raw/lo_18' SIZE 20480M ,
'/home/oracle/dev/raw/lo_19' SIZE 20480M ,
'/home/oracle/dev/raw/lo_20' SIZE 20480M ,
'/home/oracle/dev/raw/lo_21' SIZE 20480M ,
'/home/oracle/dev/raw/lo_22' SIZE 20480M ,
'/home/oracle/dev/raw/lo_23' SIZE 20480M ,
'/home/oracle/dev/raw/lo_24' SIZE 20480M ,
'/home/oracle/dev/raw/lo_25' SIZE 20480M ,
'/home/oracle/dev/raw/lo_26' SIZE 20480M ,
'/home/oracle/dev/raw/lo_27' SIZE 20480M ,
'/home/oracle/dev/raw/lo_28' SIZE 20480M ,
'/home/oracle/dev/raw/lo_29' SIZE 20480M ,
'/home/oracle/dev/raw/lo_30' SIZE 20480M ,
'/home/oracle/dev/raw/lo_31' SIZE 20480M ,
'/home/oracle/dev/raw/lo_32' SIZE 20480M;
CREATE DISKGROUP TEMP External REDUNDANCY  DISK
'/home/oracle/dev/raw/t_1' SIZE 10240M ,
'/home/oracle/dev/raw/t_2' SIZE 10240M ,
'/home/oracle/dev/raw/t_3' SIZE 10240M ,
'/home/oracle/dev/raw/t_4' SIZE 10240M ,
'/home/oracle/dev/raw/t_5' SIZE 10240M ,
'/home/oracle/dev/raw/t_6' SIZE 10240M ,
'/home/oracle/dev/raw/t_7' SIZE 10240M ,
'/home/oracle/dev/raw/t_8' SIZE 10240M ,
'/home/oracle/dev/raw/t_9' SIZE 10240M ,
'/home/oracle/dev/raw/t_10' SIZE 10240M ,
'/home/oracle/dev/raw/t_11' SIZE 10240M ,
'/home/oracle/dev/raw/t_12' SIZE 10240M ,
'/home/oracle/dev/raw/t_13' SIZE 10240M ,
'/home/oracle/dev/raw/t_14' SIZE 10240M ,
'/home/oracle/dev/raw/t_15' SIZE 10240M ,
'/home/oracle/dev/raw/t_16' SIZE 10240M ,
'/home/oracle/dev/raw/t_17' SIZE 10240M ,
'/home/oracle/dev/raw/t_18' SIZE 10240M ,
'/home/oracle/dev/raw/t_19' SIZE 10240M ,
'/home/oracle/dev/raw/t_20' SIZE 10240M ,
'/home/oracle/dev/raw/t_21' SIZE 10240M ,
'/home/oracle/dev/raw/t_22' SIZE 10240M ,
'/home/oracle/dev/raw/t_23' SIZE 10240M ,
'/home/oracle/dev/raw/t_24' SIZE 10240M ,
'/home/oracle/dev/raw/t_25' SIZE 10240M ,
'/home/oracle/dev/raw/t_26' SIZE 10240M ,
'/home/oracle/dev/raw/t_27' SIZE 10240M ,
'/home/oracle/dev/raw/t_28' SIZE 10240M ,
'/home/oracle/dev/raw/t_29' SIZE 10240M ,
'/home/oracle/dev/raw/t_30' SIZE 10240M ,
'/home/oracle/dev/raw/t_31' SIZE 10240M ,
'/home/oracle/dev/raw/t_32' SIZE 10240M;

!



---------------------------------------------
dapop_10gR2.sh
---------------------------------------------
#!/bin/bash
sqlplus /NOLOG <<EOF
connect / as sysdba
drop user tpch cascade;
grant DBA
to tpch identified by tpch;
connect tpch/tpch;
drop directory ff1;
drop directory ff2;
drop directory ff3;
drop directory ff4;
drop directory ff5;
drop directory ff6;
drop directory ff7;
drop directory ff8;
drop directory ff9;
drop directory ff10;
drop directory ff11;
drop directory ff12;
drop directory ff13;
drop directory ff14;
drop directory ff15;
drop directory ff16;
drop directory ff17;
drop directory ff18;
drop directory ff19;
drop directory ff20;
drop directory ff21;
drop directory ff22;
drop directory ff23;
drop directory ff24;
drop directory ff25;
drop directory ff26;
drop directory ff27;
drop directory ff28;
drop directory ff29;
drop directory ff30;
drop directory ff31;
drop directory ff32;

create directory ff1 as '/home/oracle/dev/ff_1';
create directory ff2 as '/home/oracle/dev/ff_2';
create directory ff3 as '/home/oracle/dev/ff_3';
create directory ff4 as '/home/oracle/dev/ff_4';
create directory ff5 as '/home/oracle/dev/ff_5';
create directory ff6 as '/home/oracle/dev/ff_6';
create directory ff7 as '/home/oracle/dev/ff_7';
create directory ff8 as '/home/oracle/dev/ff_8';
create directory ff9 as '/home/oracle/dev/ff_9';
create directory ff10 as '/home/oracle/dev/ff_10';
create directory ff11 as '/home/oracle/dev/ff_11';
create directory ff12 as '/home/oracle/dev/ff_12';
create directory ff13 as '/home/oracle/dev/ff_13';
create directory ff14 as '/home/oracle/dev/ff_14';
create directory ff15 as '/home/oracle/dev/ff_15';
create directory ff16 as '/home/oracle/dev/ff_16';
create directory ff17 as '/home/oracle/dev/ff_17';
create directory ff18 as '/home/oracle/dev/ff_18';
create directory ff19 as '/home/oracle/dev/ff_19';
create directory ff20 as '/home/oracle/dev/ff_20';
create directory ff21 as '/home/oracle/dev/ff_21';
create directory ff22 as '/home/oracle/dev/ff_22';
create directory ff23 as '/home/oracle/dev/ff_23';
create directory ff24 as '/home/oracle/dev/ff_24';
create directory ff25 as '/home/oracle/dev/ff_25';
create directory ff26 as '/home/oracle/dev/ff_26';
create directory ff27 as '/home/oracle/dev/ff_27';
create directory ff28 as '/home/oracle/dev/ff_28';
create directory ff29 as '/home/oracle/dev/ff_29';
create directory ff30 as '/home/oracle/dev/ff_30';
create directory ff31 as '/home/oracle/dev/ff_31';
create directory ff32 as '/home/oracle/dev/ff_32';



drop table l_et;
create table l_et(
   l_orderkey          number ,
   l_partkey           number ,
   l_suppkey           number ,
   l_linenumber        number ,
   l_quantity          number ,
   l_extendedprice     number ,
   l_discount          number ,
   l_tax               number ,
   l_returnflag        char(1) ,
   l_linestatus        char(1) ,
   l_shipdate          date ,
```

```
    l_commitdate        date ,
    l_receiptdate       date ,
    l_shipinstruct      char(25) ,
    l_shipmode          char(10) ,
    l_comment           varchar(44)
)
organization external (
type ORACLE_LOADER
default directory ff1
access parameters
(
                                records delimited by newline
                                nobadfile
                                nologfile
                                fields terminated by '|'
                                missing field values are null
                )
                location (
ff1:'lineitem.tbl.1',
ff2:'lineitem.tbl.2',
ff3:'lineitem.tbl.3',
ff4:'lineitem.tbl.4',
ff5:'lineitem.tbl.5',
ff6:'lineitem.tbl.6',
ff7:'lineitem.tbl.7',
ff8:'lineitem.tbl.8',
ff9:'lineitem.tbl.9',
ff10:'lineitem.tbl.10',
ff11:'lineitem.tbl.11',
ff12:'lineitem.tbl.12',
ff13:'lineitem.tbl.13',
ff14:'lineitem.tbl.14',
ff15:'lineitem.tbl.15',
ff16:'lineitem.tbl.16',
ff17:'lineitem.tbl.17',
ff18:'lineitem.tbl.18',
ff19:'lineitem.tbl.19',
ff20:'lineitem.tbl.20',
ff21:'lineitem.tbl.21',
ff22:'lineitem.tbl.22',
ff23:'lineitem.tbl.23',
ff24:'lineitem.tbl.24',
ff25:'lineitem.tbl.25',
ff26:'lineitem.tbl.26',
ff27:'lineitem.tbl.27',
ff28:'lineitem.tbl.28',
ff29:'lineitem.tbl.29',
ff30:'lineitem.tbl.30',
ff31:'lineitem.tbl.31',
ff32:'lineitem.tbl.32'
))
reject limit unlimited;

drop table o_et;
create table o_et(
    o_orderkey          number ,
    o_custkey           number ,
    o_orderstatus       char(1) ,
    o_totalprice        number ,
    o_orderdate         date ,
    o_orderpriority     char(15) ,
    o_clerk             char(15) ,
    o_shippriority      number ,
    o_comment           varchar(79)
)
organization external (
type ORACLE_LOADER
default directory ff1
access parameters
(
                                records delimited by newline
                                nobadfile
                                nologfile
                                fields terminated by '|'
                                missing field values are null
                )
                location (
ff1:'orders.tbl.1',
ff2:'orders.tbl.2',
ff3:'orders.tbl.3',
ff4:'orders.tbl.4',
ff5:'orders.tbl.5',
ff6:'orders.tbl.6',
ff7:'orders.tbl.7',
ff8:'orders.tbl.8',
ff9:'orders.tbl.9',
ff10:'orders.tbl.10',
ff11:'orders.tbl.11',
ff12:'orders.tbl.12',
ff13:'orders.tbl.13',
ff14:'orders.tbl.14',
ff15:'orders.tbl.15',
ff16:'orders.tbl.16',
ff17:'orders.tbl.17',
ff18:'orders.tbl.18',
ff19:'orders.tbl.19',
ff20:'orders.tbl.20',
ff21:'orders.tbl.21',
ff22:'orders.tbl.22',
ff23:'orders.tbl.23',
ff24:'orders.tbl.24',
ff25:'orders.tbl.25',
ff26:'orders.tbl.26',
ff27:'orders.tbl.27',
ff28:'orders.tbl.28',
ff29:'orders.tbl.29',
ff30:'orders.tbl.30',
ff31:'orders.tbl.31',
ff32:'orders.tbl.32'
))
reject limit unlimited;

drop table ps_et;
create table ps_et(
    ps_partkey          number ,
    ps_suppkey          number ,
    ps_availqty         number ,
    ps_supplycost       number ,
    ps_comment          varchar(199)
)
organization external (
type ORACLE_LOADER
default directory ff1
access parameters
(
                                records delimited by newline
                                nobadfile
                                nologfile
                                fields terminated by '|'
                                missing field values are null
                )
                location (
ff1:'partsupp.tbl.1',
ff2:'partsupp.tbl.2',
ff3:'partsupp.tbl.3',
ff4:'partsupp.tbl.4',
ff5:'partsupp.tbl.5',
ff6:'partsupp.tbl.6',
ff7:'partsupp.tbl.7',
ff8:'partsupp.tbl.8',
ff9:'partsupp.tbl.9',
ff10:'partsupp.tbl.10',
ff11:'partsupp.tbl.11',
ff12:'partsupp.tbl.12',
ff13:'partsupp.tbl.13',
ff14:'partsupp.tbl.14',
ff15:'partsupp.tbl.15',
ff16:'partsupp.tbl.16',
ff17:'partsupp.tbl.17',
ff18:'partsupp.tbl.18',
ff19:'partsupp.tbl.19',
ff20:'partsupp.tbl.20',
ff21:'partsupp.tbl.21',
ff22:'partsupp.tbl.22',
ff23:'partsupp.tbl.23',
ff24:'partsupp.tbl.24',
ff25:'partsupp.tbl.25',
ff26:'partsupp.tbl.26',
ff27:'partsupp.tbl.27',
ff28:'partsupp.tbl.28',
ff29:'partsupp.tbl.29',
ff30:'partsupp.tbl.30',
ff31:'partsupp.tbl.31',
ff32:'partsupp.tbl.32'
))
reject limit unlimited;

drop table p_et;
create table p_et(
    p_partkey           number ,
    p_name              varchar(55) ,
    p_mfgr              char(25) ,
    p_brand             char(10) ,
    p_type              varchar(25) ,
    p_size              number ,
    p_container         char(10) ,
    p_retailprice       number ,
    p_comment           varchar(23)
)
organization external (
type ORACLE_LOADER
default directory ff1
access parameters
(
                                records delimited by newline
                                nobadfile
```

```
                        nologfile
                        fields terminated by '|'
                        missing field values are null
                )
                location (
ff1:'part.tbl.1',
ff2:'part.tbl.2',
ff3:'part.tbl.3',
ff4:'part.tbl.4',
ff5:'part.tbl.5',
ff6:'part.tbl.6',
ff7:'part.tbl.7',
ff8:'part.tbl.8',
ff9:'part.tbl.9',
ff10:'part.tbl.10',
ff11:'part.tbl.11',
ff12:'part.tbl.12',
ff13:'part.tbl.13',
ff14:'part.tbl.14',
ff15:'part.tbl.15',
ff16:'part.tbl.16',
ff17:'part.tbl.17',
ff18:'part.tbl.18',
ff19:'part.tbl.19',
ff20:'part.tbl.20',
ff21:'part.tbl.21',
ff22:'part.tbl.22',
ff23:'part.tbl.23',
ff24:'part.tbl.24',
ff25:'part.tbl.25',
ff26:'part.tbl.26',
ff27:'part.tbl.27',
ff28:'part.tbl.28',
ff29:'part.tbl.29',
ff30:'part.tbl.30',
ff31:'part.tbl.31',
ff32:'part.tbl.32'
))
reject limit unlimited;

drop table c_et;
create table c_et(
   c_custkey        number ,
   c_name           varchar(25) ,
   c_address        varchar(40) ,
   c_nationkey      number ,
   c_phone          char(15) ,
   c_acctbal        number ,
   c_mktsegment     char(10) ,
   c_comment        varchar(117)
)
organization external (
type ORACLE_LOADER
default directory ff1
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
                )
                location (
ff1:'customer.tbl.1',
ff2:'customer.tbl.2',
ff3:'customer.tbl.3',
ff4:'customer.tbl.4',
ff5:'customer.tbl.5',
ff6:'customer.tbl.6',
ff7:'customer.tbl.7',
ff8:'customer.tbl.8',
ff9:'customer.tbl.9',
ff10:'customer.tbl.10',
ff11:'customer.tbl.11',
ff12:'customer.tbl.12',
ff13:'customer.tbl.13',
ff14:'customer.tbl.14',
ff15:'customer.tbl.15',
ff16:'customer.tbl.16',
ff17:'customer.tbl.17',
ff18:'customer.tbl.18',
ff19:'customer.tbl.19',
ff20:'customer.tbl.20',
ff21:'customer.tbl.21',
ff22:'customer.tbl.22',
ff23:'customer.tbl.23',
ff24:'customer.tbl.24',
ff25:'customer.tbl.25',
ff26:'customer.tbl.26',
ff27:'customer.tbl.27',
ff28:'customer.tbl.28',
ff29:'customer.tbl.29',
ff30:'customer.tbl.30',
ff31:'customer.tbl.31',
ff32:'customer.tbl.32'
))
reject limit unlimited;

drop table s_et;
create table s_et(
   s_suppkey        number ,
   s_name           char(25) ,
   s_address        varchar(40) ,
   s_nationkey      number ,
   s_phone          char(15) ,
   s_acctbal        number ,
   s_comment        varchar(101)
)
organization external (
type ORACLE_LOADER
default directory ff1
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
                )
                location (
ff1:'supplier.tbl.1',
ff2:'supplier.tbl.2',
ff3:'supplier.tbl.3',
ff4:'supplier.tbl.4',
ff5:'supplier.tbl.5',
ff6:'supplier.tbl.6',
ff7:'supplier.tbl.7',
ff8:'supplier.tbl.8',
ff9:'supplier.tbl.9',
ff10:'supplier.tbl.10',
ff11:'supplier.tbl.11',
ff12:'supplier.tbl.12',
ff13:'supplier.tbl.13',
ff14:'supplier.tbl.14',
ff15:'supplier.tbl.15',
ff16:'supplier.tbl.16',
ff17:'supplier.tbl.17',
ff18:'supplier.tbl.18',
ff19:'supplier.tbl.19',
ff20:'supplier.tbl.20',
ff21:'supplier.tbl.21',
ff22:'supplier.tbl.22',
ff23:'supplier.tbl.23',
ff24:'supplier.tbl.24',
ff25:'supplier.tbl.25',
ff26:'supplier.tbl.26',
ff27:'supplier.tbl.27',
ff28:'supplier.tbl.28',
ff29:'supplier.tbl.29',
ff30:'supplier.tbl.30',
ff31:'supplier.tbl.31',
ff32:'supplier.tbl.32'
                ))
reject limit unlimited;

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 ff1
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
                )
                location (
ff32:'nation.tbl'
))
reject limit unlimited;

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 ff1
access parameters
(
                                records delimited by newline
                                nobadfile
                                nologfile
                                fields terminated by '|'
                                missing field values are null
                )
                location (
ff32:'region.tbl'
))
reject limit unlimited;

alter table l_et parallel;
alter table o_et parallel;
alter table ps_et parallel;
alter table p_et parallel;
alter table c_et parallel;
alter table s_et parallel;

alter user tpch default tablespace ts_default;
alter user tpch temporary tablespace ts_temp;

@?/rdbms/admin/utlxplan.sql;

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
tablespace ts_data
storage (initial 22m freelist groups 8 freelists 99)
compress
parallel
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'))
,
partition ord2 values less than (to_date('1992-02-01','YYYY-MM-DD'))
,
partition ord3 values less than (to_date('1992-03-01','YYYY-MM-DD'))
,
partition ord4 values less than (to_date('1992-04-01','YYYY-MM-DD'))
,
partition ord5 values less than (to_date('1992-05-01','YYYY-MM-DD'))
,
partition ord6 values less than (to_date('1992-06-01','YYYY-MM-DD'))
,
partition ord7 values less than (to_date('1992-07-01','YYYY-MM-DD'))
,
partition ord8 values less than (to_date('1992-08-01','YYYY-MM-DD'))
,
partition ord9 values less than (to_date('1992-09-01','YYYY-MM-DD'))
,
partition ord10 values less than (to_date('1992-10-01','YYYY-MM-DD'))
,
partition ord11 values less than (to_date('1992-11-01','YYYY-MM-DD'))
,
partition ord12 values less than (to_date('1992-12-01','YYYY-MM-DD'))
,
partition ord13 values less than (to_date('1993-01-01','YYYY-MM-DD'))
,
partition ord14 values less than (to_date('1993-02-01','YYYY-MM-DD'))
,
partition ord15 values less than (to_date('1993-03-01','YYYY-MM-DD'))
,
partition ord16 values less than (to_date('1993-04-01','YYYY-MM-DD'))
,
partition ord17 values less than (to_date('1993-05-01','YYYY-MM-DD'))
,
partition ord18 values less than (to_date('1993-06-01','YYYY-MM-DD'))
,
partition ord19 values less than (to_date('1993-07-01','YYYY-MM-DD'))
,
partition ord20 values less than (to_date('1993-08-01','YYYY-MM-DD'))
,
partition ord21 values less than (to_date('1993-09-01','YYYY-MM-DD'))
,
partition ord22 values less than (to_date('1993-10-01','YYYY-MM-DD'))
,
partition ord23 values less than (to_date('1993-11-01','YYYY-MM-DD'))
,
partition ord24 values less than (to_date('1993-12-01','YYYY-MM-DD'))
,
partition ord25 values less than (to_date('1994-01-01','YYYY-MM-DD'))
,
partition ord26 values less than (to_date('1994-02-01','YYYY-MM-DD'))
,
partition ord27 values less than (to_date('1994-03-01','YYYY-MM-DD'))
,
partition ord28 values less than (to_date('1994-04-01','YYYY-MM-DD'))
,
partition ord29 values less than (to_date('1994-05-01','YYYY-MM-DD'))
,
partition ord30 values less than (to_date('1994-06-01','YYYY-MM-DD'))
,
partition ord31 values less than (to_date('1994-07-01','YYYY-MM-DD'))
,
partition ord32 values less than (to_date('1994-08-01','YYYY-MM-DD'))
,
partition ord33 values less than (to_date('1994-09-01','YYYY-MM-DD'))
,
partition ord34 values less than (to_date('1994-10-01','YYYY-MM-DD'))
,
partition ord35 values less than (to_date('1994-11-01','YYYY-MM-DD'))
,
partition ord36 values less than (to_date('1994-12-01','YYYY-MM-DD'))
,
partition ord37 values less than (to_date('1995-01-01','YYYY-MM-DD'))
,
partition ord38 values less than (to_date('1995-02-01','YYYY-MM-DD'))
,
partition ord39 values less than (to_date('1995-03-01','YYYY-MM-DD'))
,
partition ord40 values less than (to_date('1995-04-01','YYYY-MM-DD'))
,
partition ord41 values less than (to_date('1995-05-01','YYYY-MM-DD'))
,
partition ord42 values less than (to_date('1995-06-01','YYYY-MM-DD'))
,
partition ord43 values less than (to_date('1995-07-01','YYYY-MM-DD'))
,
partition ord44 values less than (to_date('1995-08-01','YYYY-MM-DD'))
,
partition ord45 values less than (to_date('1995-09-01','YYYY-MM-DD'))
,
partition ord46 values less than (to_date('1995-10-01','YYYY-MM-DD'))
,
partition ord47 values less than (to_date('1995-11-01','YYYY-MM-DD'))
,
partition ord48 values less than (to_date('1995-12-01','YYYY-MM-DD'))
,
partition ord49 values less than (to_date('1996-01-01','YYYY-MM-DD'))
,
partition ord50 values less than (to_date('1996-02-01','YYYY-MM-DD'))
,
partition ord51 values less than (to_date('1996-03-01','YYYY-MM-DD'))
,
partition ord52 values less than (to_date('1996-04-01','YYYY-MM-DD'))
,
partition ord53 values less than (to_date('1996-05-01','YYYY-MM-DD'))
,
partition ord54 values less than (to_date('1996-06-01','YYYY-MM-DD'))
,
partition ord55 values less than (to_date('1996-07-01','YYYY-MM-DD'))
,
partition ord56 values less than (to_date('1996-08-01','YYYY-MM-DD'))
,
partition ord57 values less than (to_date('1996-09-01','YYYY-MM-DD'))
,
partition ord58 values less than (to_date('1996-10-01','YYYY-MM-DD'))
,
partition ord59 values less than (to_date('1996-11-01','YYYY-MM-DD'))
,
partition ord60 values less than (to_date('1996-12-01','YYYY-MM-DD'))
,
partition ord61 values less than (to_date('1997-01-01','YYYY-MM-DD'))
,
partition ord62 values less than (to_date('1997-02-01','YYYY-MM-DD'))
,
partition ord63 values less than (to_date('1997-03-01','YYYY-MM-DD'))
,
partition ord64 values less than (to_date('1997-04-01','YYYY-MM-DD'))
,
partition ord65 values less than (to_date('1997-05-01','YYYY-MM-DD'))
,
partition ord66 values less than (to_date('1997-06-01','YYYY-MM-DD'))
,
partition ord67 values less than (to_date('1997-07-01','YYYY-MM-DD'))
```

```
,
partition ord68 values less than (to_date('1997-08-01','YYYY-MM-DD'))
,
partition ord69 values less than (to_date('1997-09-01','YYYY-MM-DD'))
,
partition ord70 values less than (to_date('1997-10-01','YYYY-MM-DD'))
,
partition ord71 values less than (to_date('1997-11-01','YYYY-MM-DD'))
,
partition ord72 values less than (to_date('1997-12-01','YYYY-MM-DD'))
,
partition ord73 values less than (to_date('1998-01-01','YYYY-MM-DD'))
,
partition ord74 values less than (to_date('1998-02-01','YYYY-MM-DD'))
,
partition ord75 values less than (to_date('1998-03-01','YYYY-MM-DD'))
,
partition ord76 values less than (to_date('1998-04-01','YYYY-MM-DD'))
,
partition ord77 values less than (to_date('1998-05-01','YYYY-MM-DD'))
,
partition ord78 values less than (to_date('1998-06-01','YYYY-MM-DD'))
,
partition ord79 values less than (to_date('1998-07-01','YYYY-MM-DD'))
,
partition ord80 values less than (to_date('1998-08-01','YYYY-MM-DD'))
,
partition ord81 values less than (to_date('1998-09-01','YYYY-MM-DD'))
,
partition ord82 values less than (to_date('1998-10-01','YYYY-MM-DD'))
,
partition ord83 values less than (to_date('1998-11-01','YYYY-MM-DD'))
,
partition ord84 values less than (MAXVALUE)
)
as select
   o_orderdate         ,
   o_orderkey          ,
   o_custkey           ,
   o_orderpriority     ,
   o_shippriority      ,
   o_clerk             ,
   o_orderstatus       ,
   o_totalprice        ,
   o_comment
from o_et;
!date

rem drop table partsupp;
create table partsupp(
   ps_partkey          NOT NULL,
   ps_suppkey          NOT NULL,
   ps_supplycost       NOT NULL,
   ps_availqty         ,
   ps_comment
)
partition by hash(ps_partkey)
partitions 16
storage (initial 200m freelist groups 8 freelists 99)
parallel
nologging
tablespace ts_data
as select
   ps_partkey          ,
   ps_suppkey          ,
   ps_supplycost       ,
   ps_availqty         ,
   ps_comment
from ps_et;
!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
storage (initial 100m freelist groups 8 freelists 99)
compress
parallel
nologging
partition by hash (c_custkey)
partitions 16
tablespace ts_data
as select
   c_custkey           ,
   c_mktsegment        ,
   c_nationkey         ,
   c_name              ,
   c_address           ,
   c_phone             ,
   c_acctbal           ,
   c_comment
from c_et;
!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
storage (initial 132m  freelist groups 8 freelists 99)
compress
parallel
nologging
partition by hash (p_partkey)
partitions 16
tablespace ts_data
as select
   p_partkey           ,
   p_type              ,
   p_size              ,
   p_brand             ,
   p_name              ,
   p_container         ,
   p_mfgr              ,
   p_retailprice       ,
   p_comment
from p_et;
!date

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
storage (initial 16m freelist groups 8 freelists 99)
compress
parallel
nologging
partition by hash (s_suppkey)
partitions 16
tablespace ts_data
as select
   s_suppkey           ,
   s_nationkey         ,
   s_comment           ,
   s_name              ,
   s_address           ,
   s_phone             ,
   s_acctbal
from s_et;
!date

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

rem drop table region;
create table region(
   r_regionkey         ,
   r_name              ,
   r_comment           )
as select * from r_et;
!date

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
tablespace ts_data
storage (initial 100m freelist groups 8 freelists 99)
compress
parallel
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')) ,
partition item2 values less than (to_date('1992-02-01','YYYY-MM-DD')) ,
partition item3 values less than (to_date('1992-03-01','YYYY-MM-DD')) ,
partition item4 values less than (to_date('1992-04-01','YYYY-MM-DD')) ,
partition item5 values less than (to_date('1992-05-01','YYYY-MM-DD')) ,
partition item6 values less than (to_date('1992-06-01','YYYY-MM-DD')) ,
partition item7 values less than (to_date('1992-07-01','YYYY-MM-DD')) ,
partition item8 values less than (to_date('1992-08-01','YYYY-MM-DD')) ,
partition item9 values less than (to_date('1992-09-01','YYYY-MM-DD')) ,
partition item10 values less than (to_date('1992-10-01','YYYY-MM-DD')) ,
partition item11 values less than (to_date('1992-11-01','YYYY-MM-DD')) ,
partition item12 values less than (to_date('1992-12-01','YYYY-MM-DD')) ,
partition item13 values less than (to_date('1993-01-01','YYYY-MM-DD')) ,
partition item14 values less than (to_date('1993-02-01','YYYY-MM-DD')) ,
partition item15 values less than (to_date('1993-03-01','YYYY-MM-DD')) ,
partition item16 values less than (to_date('1993-04-01','YYYY-MM-DD')) ,
partition item17 values less than (to_date('1993-05-01','YYYY-MM-DD')) ,
partition item18 values less than (to_date('1993-06-01','YYYY-MM-DD')) ,
partition item19 values less than (to_date('1993-07-01','YYYY-MM-DD')) ,
partition item20 values less than (to_date('1993-08-01','YYYY-MM-DD')) ,
partition item21 values less than (to_date('1993-09-01','YYYY-MM-DD')) ,
partition item22 values less than (to_date('1993-10-01','YYYY-MM-DD')) ,
partition item23 values less than (to_date('1993-11-01','YYYY-MM-DD')) ,
partition item24 values less than (to_date('1993-12-01','YYYY-MM-DD')) ,
partition item25 values less than (to_date('1994-01-01','YYYY-MM-DD')) ,
partition item26 values less than (to_date('1994-02-01','YYYY-MM-DD')) ,
partition item27 values less than (to_date('1994-03-01','YYYY-MM-DD')) ,
partition item28 values less than (to_date('1994-04-01','YYYY-MM-DD')) ,
partition item29 values less than (to_date('1994-05-01','YYYY-MM-DD')) ,
partition item30 values less than (to_date('1994-06-01','YYYY-MM-DD')) ,
partition item31 values less than (to_date('1994-07-01','YYYY-MM-DD')) ,
partition item32 values less than (to_date('1994-08-01','YYYY-MM-DD')) ,
partition item33 values less than (to_date('1994-09-01','YYYY-MM-DD')) ,
partition item34 values less than (to_date('1994-10-01','YYYY-MM-DD')) ,
partition item35 values less than (to_date('1994-11-01','YYYY-MM-DD')) ,
partition item36 values less than (to_date('1994-12-01','YYYY-MM-DD')) ,
partition item37 values less than (to_date('1995-01-01','YYYY-MM-DD')) ,
partition item38 values less than (to_date('1995-02-01','YYYY-MM-DD')) ,
partition item39 values less than (to_date('1995-03-01','YYYY-MM-DD')) ,
partition item40 values less than (to_date('1995-04-01','YYYY-MM-DD')) ,
partition item41 values less than (to_date('1995-05-01','YYYY-MM-DD')) ,
partition item42 values less than (to_date('1995-06-01','YYYY-MM-DD')) ,
partition item43 values less than (to_date('1995-07-01','YYYY-MM-DD')) ,
partition item44 values less than (to_date('1995-08-01','YYYY-MM-DD')) ,
partition item45 values less than (to_date('1995-09-01','YYYY-MM-DD')) ,
partition item46 values less than (to_date('1995-10-01','YYYY-MM-DD')) ,
partition item47 values less than (to_date('1995-11-01','YYYY-MM-DD')) ,
partition item48 values less than (to_date('1995-12-01','YYYY-MM-DD')) ,
partition item49 values less than (to_date('1996-01-01','YYYY-MM-DD')) ,
partition item50 values less than (to_date('1996-02-01','YYYY-MM-DD')) ,
partition item51 values less than (to_date('1996-03-01','YYYY-MM-DD')) ,
partition item52 values less than (to_date('1996-04-01','YYYY-MM-DD')) ,
partition item53 values less than (to_date('1996-05-01','YYYY-MM-DD')) ,
partition item54 values less than (to_date('1996-06-01','YYYY-MM-DD')) ,
partition item55 values less than (to_date('1996-07-01','YYYY-MM-DD')) ,
partition item56 values less than (to_date('1996-08-01','YYYY-MM-DD')) ,
partition item57 values less than (to_date('1996-09-01','YYYY-MM-DD')) ,
partition item58 values less than (to_date('1996-10-01','YYYY-MM-DD')) ,
partition item59 values less than (to_date('1996-11-01','YYYY-MM-DD')) ,
partition item60 values less than (to_date('1996-12-01','YYYY-MM-DD')) ,
partition item61 values less than (to_date('1997-01-01','YYYY-MM-DD')) ,
partition item62 values less than (to_date('1997-02-01','YYYY-MM-DD')) ,
partition item63 values less than (to_date('1997-03-01','YYYY-MM-DD')) ,
partition item64 values less than (to_date('1997-04-01','YYYY-MM-DD')) ,
partition item65 values less than (to_date('1997-05-01','YYYY-MM-DD')) ,
partition item66 values less than (to_date('1997-06-01','YYYY-MM-DD')) ,
partition item67 values less than (to_date('1997-07-01','YYYY-MM-DD')) ,
partition item68 values less than (to_date('1997-08-01','YYYY-MM-DD')) ,
partition item69 values less than (to_date('1997-09-01','YYYY-MM-DD')) ,
partition item70 values less than (to_date('1997-10-01','YYYY-MM-DD')) ,
partition item71 values less than (to_date('1997-11-01','YYYY-MM-DD')) ,
partition item72 values less than (to_date('1997-12-01','YYYY-MM-DD')) ,
partition item73 values less than (to_date('1998-01-01','YYYY-MM-DD')) ,
partition item74 values less than (to_date('1998-02-01','YYYY-MM-DD')) ,
partition item75 values less than (to_date('1998-03-01','YYYY-MM-DD')) ,
partition item76 values less than (to_date('1998-04-01','YYYY-MM-DD')) ,
partition item77 values less than (to_date('1998-05-01','YYYY-MM-DD')) ,
partition item78 values less than (to_date('1998-06-01','YYYY-MM-DD')) ,
partition item79 values less than (to_date('1998-07-01','YYYY-MM-DD')) ,
partition item80 values less than (to_date('1998-08-01','YYYY-MM-DD')) ,
partition item81 values less than (to_date('1998-09-01','YYYY-MM-DD')) ,
partition item82 values less than (to_date('1998-10-01','YYYY-MM-DD')) ,
partition item83 values less than (to_date('1998-11-01','YYYY-MM-DD')) ,
partition item84 values less than (MAXVALUE)
)
as select
   l_shipdate          ,
   l_orderkey          ,
   l_discount          ,
   l_extendedprice     ,
   l_suppkey           ,
   l_quantity          ,
   l_returnflag        ,
   l_partkey           ,
   l_linestatus        ,
   l_tax               ,
   l_commitdate        ,
   l_receiptdate       ,
   l_shipmode          ,
   l_linenumber        ,
   l_shipinstruct      ,
   l_comment
from l_et;
!date

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;

!date

rem drop index i_l_orderkey;
create index i_l_orderkey
on lineitem (l_orderkey) global partition by hash (l_orderkey)
partitions 16
pctfree 2
initrans 10
tablespace ts_data
storage (initial 132m freelist groups 8 freelists 99)
parallel
compute statistics
nologging;
alter index i_l_orderkey allocate extent (size 132m);
alter index i_l_orderkey allocate extent (size 132m);
alter index i_l_orderkey allocate extent (size 132m);
alter index i_l_orderkey allocate extent (size 132m);

!date

rem drop index i_o_orderkey;
create unique index i_o_orderkey
on orders (o_orderkey) global partition by hash (o_orderkey)
partitions 16
pctfree 2
initrans 10
tablespace ts_data
storage (initial 16m freelist groups 8 freelists 99 )
parallel
compute statistics
nologging;
alter index i_o_orderkey allocate extent (size 16m);
alter index i_o_orderkey allocate extent (size 16m);
alter index i_o_orderkey allocate extent (size 16m);
alter index i_o_orderkey allocate extent (size 16m);

!date

rem drop index i_c_custkey;
create unique index i_c_custkey
on customer (c_custkey)  global partition by hash (c_custkey)
partitions 16
pctfree 2
initrans 10
```

```
tablespace ts_data
storage (initial 10m freelist groups 8 freelists 99)
parallel
compute statistics
nologging;

create unique index i_ps_partkey_suppkey
on partsupp (ps_partkey,ps_suppkey)  global partition by hash (ps_partkey)
partitions 16
pctfree 2
initrans 10
tablespace ts_data
storage (initial 10m freelist groups 8 freelists 99)
parallel
compute statistics
nologging;

!date
set timing on
execute dbms_stats.gather_schema_stats('TPCH' , estimate_percent => 1, degree => 16 ,
granularity => 'GLOBAL' );
connect / as sysdba
execute dbms_stats.gather_system_stats;
exec dbms_scheduler.disable('GATHER_STATS_JOB');
exec dbms_scheduler.disable('AUTO_SPACE_ADVISOR_JOB');
exec dbms_scheduler.disable('AUTO_TASKS_JOB_CLASS');
alter system switch logfile;
!date
EOF


---------------------------------------------
dbcre_10gR2.sh
---------------------------------------------
#!/bin/ksh
echo "database creation"
date;

sqlplus  /NOLOG <<!
connect /as sysdba

startup pfile = /home/oracle/10gR2_rel/db/dbs/init_build.ora nomount;
create database
  controlfile reuse
  logfile group 1 ('/home/oracle/dev/raw/log_1_1a','/home/oracle/dev/raw/log_1_1b')  size 4096m
reuse,
        group 2 ('/home/oracle/dev/raw/log_1_2a','/home/oracle/dev/raw/log_1_2b')  size 4096m
reuse
  datafile '/home/oracle/dev/raw/sys'  size 512m reuse
  sysaux datafile '/home/oracle/dev/raw/sysaux-1' size 500m reuse
  undo tablespace ts_undo1
    datafile '/home/oracle/dev/raw/undo1' size 8192m reuse
  default temporary tablespace ts_temp
    tempfile '+TEMP' size 10240m reuse
    extent management local uniform size 10m
  maxdatafiles 4000
  maxinstances 8;

create undo tablespace ts_undo2 datafile '/home/oracle/dev/raw/undo2' size 8192m reuse;
create undo tablespace ts_undo3 datafile '/home/oracle/dev/raw/undo3' size 8192m reuse;
create undo tablespace ts_undo4 datafile '/home/oracle/dev/raw/undo4' size 8192m reuse;
create undo tablespace ts_undo5 datafile '/home/oracle/dev/raw/undo5' size 8192m reuse;
create undo tablespace ts_undo6 datafile '/home/oracle/dev/raw/undo6' size 8192m reuse;
create undo tablespace ts_undo7 datafile '/home/oracle/dev/raw/undo7' size 8192m reuse;
create undo tablespace ts_undo8 datafile '/home/oracle/dev/raw/undo8' size 8192m reuse;

alter database add logfile thread 2  group 3
('/home/oracle/dev/raw/log_2_1a','/home/oracle/dev/raw/log_2_1b') size 4096m reuse;
alter database add logfile thread 2  group 4
('/home/oracle/dev/raw/log_2_2a','/home/oracle/dev/raw/log_2_2b') size 4096m reuse;
alter database add logfile thread 3  group 5 ('/home/oracle/dev/raw/log_3_1a',
'/home/oracle/dev/raw/log_3_1b') size 4096m reuse;
alter database add logfile thread 3  group 6 ('/home/oracle/dev/raw/log_3_2a',
'/home/oracle/dev/raw/log_3_2b') size 4096m reuse;
alter database add logfile thread 4  group 7 ('/home/oracle/dev/raw/log_4_1a',
'/home/oracle/dev/raw/log_4_1b') size 4096m reuse;
alter database add logfile thread 4  group 8 ('/home/oracle/dev/raw/log_4_2a',
'/home/oracle/dev/raw/log_4_2b') size 4096m reuse;
alter database add logfile thread 5  group 9 ('/home/oracle/dev/raw/log_5_1a',
'/home/oracle/dev/raw/log_5_1b') size 4096m reuse;
alter database add logfile thread 5  group 10 ('/home/oracle/dev/raw/log_5_2a',
'/home/oracle/dev/raw/log_5_2b') size 4096m reuse;
alter database add logfile thread 6  group 11 ('/home/oracle/dev/raw/log_6_1a',
'/home/oracle/dev/raw/log_6_1b') size 4096m reuse;
alter database add logfile thread 6  group 12 ('/home/oracle/dev/raw/log_6_2a',
'/home/oracle/dev/raw/log_6_2b') size 4096m reuse;
alter database add logfile thread 7  group 13 ('/home/oracle/dev/raw/log_7_1a',
'/home/oracle/dev/raw/log_7_1b') size 4096m reuse;
alter database add logfile thread 7  group 14 ('/home/oracle/dev/raw/log_7_2a',
'/home/oracle/dev/raw/log_7_2b') size 4096m reuse;
alter database add logfile thread 8  group 15 ('/home/oracle/dev/raw/log_8_1a',
'/home/oracle/dev/raw/log_8_1b') size 4096m reuse;
alter database add logfile thread 8  group 16 ('/home/oracle/dev/raw/log_8_2a',
'/home/oracle/dev/raw/log_8_2b') size 4096m reuse;

alter database enable public thread 2;
alter database enable public thread 3;
alter database enable public thread 4;
alter database enable public thread 5;
alter database enable public thread 6;
alter database enable public thread 7;
alter database enable public thread 8;

set termout off
set echo off
spool /tmp/cat
@?/rdbms/admin/catalog.sql;
@?/rdbms/admin/catproc.sql;
@?/rdbms/admin/catclust.sql;
connect system/manager
@?/sqlplus/admin/pupbld.sql;
spool off
!
echo "end of database creation"
date


---------------------------------------------
tscre_10gR2.sh
---------------------------------------------
#!/bin/ksh
echo "START: tablespace creation"
date;

sqlplus  /NOLOG <<!
connect /as sysdba

create tablespace ts_default
datafile '/home/oracle/dev/raw/default' size 512m reuse
extent management local autoallocate;
create tablespace ts_data nologging
datafile '+DATA' size 20480m reuse extent management local autoallocate;
!
wait;

i=1
while [ $i -lt 32 ]
do
        i=`expr $i + 1`
        addts.sh  ts_data +DATA 20480m  &
done

wait;


i=1
while [ $i -lt 32 ]
do
        i=`expr $i + 1`
        addts.sh  ts_temp +TEMP 10240m &
done

wait;


echo "END: tablespace  creation"
date;
```

Appendix C: ACID Scripts

```
--------------------------------------------
a_query2.sql
--------------------------------------------
Rem
Rem $Header: aquery2.sql 07-aug-99.23:54:47 mpoess Exp $
Rem
Rem aquery2.sql
Rem
Rem  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
Rem
Rem    NAME
Rem      aquery2.sql - <one-line expansion of the name>
Rem
Rem  DESCRIPTION
Rem       Performs query on PARTSUPP for TPC-D benchmark
Rem       Isolation Test 5.
Rem       Asks user to input values for ps_partkey and ps_suppkey
Rem       The range for ps_partkey is 1 to 20000
Rem       The range for ps_suppkey is 1 to 1000
Rem       A valid combination is 46 and 47
Rem  Usage:   sqlplus tpcd/tpcd @a_query2 <ps_partkey> <ps_suppkey>
Rem
Rem    MODIFIED   (MM/DD/YY)
Rem    mpoess      08/07/99 - Creation
Rem    mpoess      08/07/99 - Created
Rem
rem  DESCRIPTION
rem       Performs query on PARTSUPP for TPC-D benchmark
rem       Isolation Test 5.
rem       Asks user to input values for ps_partkey and ps_suppkey
rem       The range for ps_partkey is 1 to 20000
rem       The range for ps_suppkey is 1 to 1000
rem       A valid combination is 46 and 47

set serverout on;

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

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

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

exit;


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

set serverout on;

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

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

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

exit;


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

. $KIT_DIR/env

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

usage() {

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

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

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

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

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

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

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

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

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


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

/*

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

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

  NOTES
    <other useful comments, qualifications, etc.>

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

*/

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

#include "atranspl.h"

/* Declare error handling functions */

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

/* declarations for ORDERS */

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

/* declarations for LINEITEM */

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

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

sb2 l_npricei;

/* other declarations */

int delta = 0;
double rprice;
double cost;

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

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

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

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

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

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

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

time_t curr_time;         /* Current Time                       */

/* OCI handles */

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

/* OCI bind handles */

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

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

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

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

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

char sqlstmt[1024];

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

void usage()
{

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

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

```
  fprintf(stderr,"    s<sleep>                :Sleep Time - sleep <sleep> seconds before commit or
rollback\n\n");
  exit(-1);

}


void ACIDexit() {

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

}



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

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

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

  (void) OCITransRollback(tpcsvc,errhp,OCI_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
-----------------------------------------------
/* Copyright (c) 2001, 2002, Oracle Corporation.  All rights reserved.  */

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

  DESCRIPTION

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

*/
#ifndef ATRANSPL_H

#define ATRANSPL_H

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

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

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

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

extern int errno;

#ifndef NULL
#define NULL 0
#endif

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

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#ifndef ub1
#define ub1 unsigned char
#endif

#define UNAME_LEN 64
#define WRITE_BUF_LEN 1024
```

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

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

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

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

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

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

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

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

#define OCIbbname(stmh,bindp,errh,sqlvar,progv,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 2)"
#define PDDLTXT "alter session force parallel ddl parallel (degree 2)"
#define OICATXT "alter session set optimizer_index_cost_adj=25"

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

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

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

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

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

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

#endif /* ATRANSPL_H */


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


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

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

CREATE OR REPLACE PACKAGE BODY d_atrans
IS
PROCEDURE doatrans
(
                l_key                           IN OUT integer,
                o_key                           IN OUT integer,
                delta                           IN OUT integer,
                l_pkey          IN OUT integer,
                l_skey          IN OUT integer,
                l_quan          IN OUT integer,
                l_newquan       IN OUT integer,
                l_tax           IN OUT number,
                l_disc          IN OUT number,
                l_eprice        IN OUT number,
                l_neweprice     IN OUT number,
                o_tprice        IN OUT number,
                o_newtprice     IN OUT number,
                rprice                          IN OUT number,
                cost                            IN OUT number
)
```

```
IS
        ototal number;
        not_serializable EXCEPTION;
        PRAGMA EXCEPTION_INIT(not_serializable,-8177);
BEGIN
 LOOP BEGIN

   select o_totalprice
                   into o_tprice
                   from orders
                   where o_orderkey = o_key;

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

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

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

   update orders
      set o_totalprice = o_newtprice
      where o_orderkey = o_key;

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

   EXIT;

 EXCEPTION
   WHEN not_serializable THEN
     ROLLBACK;
 END;

 END LOOP;

END doatrans;
END;
/
exit;


---------------------------------------------
ckpt.sh
---------------------------------------------
#!/bin/ksh
#
# $Header: ckpt.sh 08-aug-99.17:37:07 mpoess Exp $
#
# ckpt.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      ckpt.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#      Usage: ckpt.sh
#      Start database checkpoint
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/08/99 - Creation
#    mpoess      08/08/99 - Creation
#

. $KIT_DIR/env

sqlplus -s /NOLOG << !

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


---------------------------------------------
consist.sh
---------------------------------------------
#!/bin/ksh
#
# $Header: consist.sh 08-aug-99.14:20:51 mpoess Exp $
#
# consist.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      consist.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#      Performs consistency tests.
#      Usage: consist.sh [-n iter] [-s number of stream] [-p prog]
#                        [-u usr/pswd] -h
#
#      Options: See usage below
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/08/99 - Creation
#    mpoess      08/08/99 - Creation
#

. $KIT_DIR/env

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

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

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

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

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

usage() {

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

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

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


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

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

```
exit 1
fi

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

i=0

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

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

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

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

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

i=0

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

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

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

wait

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

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

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

cat ${ORACLE_HOME}/rdbms/log/alert_${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
---------------------------------------------
Rem
Rem $Header: consist.sql 08-aug-99.16:59:17 mpoess Exp $
Rem
Rem consist.sql
Rem
Rem  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
Rem
Rem    NAME
Rem      consist.sql - <one-line expansion of the name>
Rem
Rem  DESCRIPTION
Rem       Verifies the consistency of TPC-D database using the
Rem       consistency condition.
Rem
Rem Usage:   sqlplus tpcd/tpcd @consist
Rem
Rem NOTE
Rem REQUIRES PACKAGES prvtotpt and dbmsotpt
rem
Rem    MODIFIED   (MM/DD/YY)
Rem    mpoess      08/08/99 - Creation
Rem    mpoess      08/08/99 - Created
Rem

set verify off
rem set termout on
rem set echo on


REM
REM Get today's date.
REM

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

set serverout on;

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


   select 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



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


set termout on;
set serverout on;
```

```
set echo on;

drop table history;

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

exit;
```

---

end_acid.sh

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

. $KIT_DIR/env

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

SHELL_PID=`cat ${DURA_DIR}/shellpid`
ITER=100
STEM=${NUM_STREAMS}
let STEM="$STEM + 1" # add one for the update stream
PROG=${ACID_DIR}/atranspl.ott
IN=${ACID_DIR}/acid_in
DURA=${DURA_DIR}/drate
OUT=${DURA_DIR}/drate
DSMPL=${DURA_DIR}/durasmpl
KEY=${DURA_DIR}/key${SHELL_PID}_
USER=tpch/tpch
TRIG=1
HCNT=duracnta

# get history count

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

# perform the consistency

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

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

---

gtime.c

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

/*

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

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

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

  INTERNAL FUNCTION(S)
    <other external functions defined - one-line descriptions>

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

  NOTES
    <other useful comments, qualifications, etc.>

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

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

# include <sys/time.h>

main ()
{

 struct timeval tv;

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

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

}

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

---

iso1.sh

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

. $KIT_DIR/env

# May need to change the following:
RSH=ssh

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

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

USER=$DATABASE_USER
PROG=atranspl
```

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

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

usage() {

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

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

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


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

# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

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

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

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

sleep 1

# start ACID transaction, Sleep for 60 second before COMMIT

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

# let's sleep 10 seconds before starting ACID query

sleep 15

# start ACID query with the same OKEY

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

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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



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

. $KIT_DIR/env

RSH=ssh

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

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

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

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

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

sleep 1

# start ACID transaction, Sleep for 30 second before ROLLBACK

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

# let's sleep 15 seconds before starting ACID query

sleep 15

# start ACID query with the same OKEY

echo "Running ACID query 15 seconds AFTER the start of ACID transaction" \
```

```
>> $TXN2FILE
echo "`date`" >> $TXN2FILE
if [ "$HOST" != "" ]
then
echo "Starting ACID query on node $HOST" >> $TXN2FILE
${RSH} -n ${HOST}  sqlplus "$USER" @$ACID_DIR/isolation/a_query $OKEY >>
$TXN2FILE
else
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >> $TXN2FILE
fi

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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


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

. $KIT_DIR/env

# May need to change the following:
RSH=ssh

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

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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


# generate key files

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

sleep 1

# start ACID transaction, Sleep for 30 second before COMMIT

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

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

sleep 15

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

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

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

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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


---------------------------------------------
iso4.sh
---------------------------------------------
#!/bin/ksh
#
# $Header: iso4.sh 04-aug-99.09:21:12 mpoess Exp $
#
# iso4.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      iso4.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#      Usage: iso4.sh [-u user/password] [-n remote_node] -h
#      Options: See usage below
#  NOTES
#       For a cross node isolation test, assume the local node is
#       one of the participating nodes.  The other node can be
#       specified by the -n option.
#       We need to make sure the remote node has access to the
#       file system on the local node.  Otherwise, we need to rcp
#       the keyfile to the remote system.
#       You need to set the environment variable TPCD_KIT_DIR
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/04/99 - Creation
#    mpoess      08/04/99 - Creation
#

. $KIT_DIR/env

# May need to change the following:
RSH=ssh

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

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

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

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

```
usage() {

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

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

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


# generate key files

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

sleep 1

# start ACID transaction, Sleep for 30 second before ROLLBACK

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

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

sleep 15

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

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

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

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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


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

. $KIT_DIR/env

# May need to change the following:
RSH=ssh
OH=$ORACLE_HOME
# ACID_DIR=$TPCD_KIT_DIR/audit is set in env
OUT_DIR=$ACID_OUT
DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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


# generate key files

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

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

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

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

sleep 1

# start ACID transaction, Sleep for 60 second before COMMIT

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

# let's sleep 5 seconds before starting PARTSUPP query

sleep 5

# First generate PS_PARTKEY and PS_SUPPKEY

PSKEY=`randpsup 1`

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

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

wait

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

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

. $KIT_DIR/env

# May need to change the following:
RSH=ssh

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
scp $KEYFILE ${HOST}:$KEYFILE

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

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

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

sleep 1

# start Query 1, use 0 as the delta

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

# sleep 2 seconds before starting ACID transaction

sleep 2

# start ACID transaction, COMMIT after one second

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

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

# start Query 17

sleep 2

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

wait

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

cat $TXN1FILE $TXN2FILE $TXN3FILE >> $ISOFILE

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


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

/*

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

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

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

#define ORDERCNT 150000.0

/* MK_SPARSE adopted from dss.h */

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

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

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

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

long l_key = 0;
```

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

/* OCI handles */

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

OCIBind *l_key_bp;
OCIBind *o_key_bp;

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

char sqlstmt[1024];


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


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

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

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

  (void) OCITransRollback(tpcsvc,errhp,OCI_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);
}


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

/*

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

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

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

*/

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

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

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

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

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


main(argc, argv)
     int argc;
     char **argv;
{
  double sf = 0.1;           /* scale factor     */
  long supp;                 /* the 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");
}



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

. $KIT_DIR/env

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

usage() {

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

ITER=600
STEM=${NUM_STREAMS}
let STEM="$STEM + 1" # add one for the update stream
SF=1
```

```
PROG=atranspl
IN=${ACID_DIR}/acid_in
DURA_DIR=$ACID_OUT/dura
OUT=$DURA_DIR/drate
DURA=$DURA_DIR/dura
KEY=${DURA_DIR}/key$$_
echo "$$" > ${DURA_DIR}/shellpid
USER=tpch/tpch
TRIG=1
HCNT=duracntb

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

# get all the options

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


echo "Starting ACID run..."

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

# Get history count before the run

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

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

wait
# perform the consistency

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

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

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

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

done

wait

echo "ACID run completed"


-------------------------------------------
```

sample.sh

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

# $1 durability output file

. $KIT_DIR/env

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



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

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

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

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

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



-------------------------------------------
```

sample.sql

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

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

exit;
```

Appendix D: Qualifucation query text and output

```
---------------------------------------------
10.log
---------------------------------------------
Begin Execution at Thu Sep  8 14:19:56 2005


-- using default substitutions


select  *  from  (
select
c_custkey,
c_name,
sum(l_extendedprice * (1 - l_discount)) as revenue,
c_acctbal,
n_name,
c_address,
c_phone,
c_comment
from
customer,
orders,
lineitem,
nation
where
c_custkey = o_custkey
and l_orderkey = o_orderkey
and o_orderdate >= to_date ('1993-10-01',  'YYYY-MM-DD')
and o_orderdate < add_months(  to_date( '1993-10-01',  'YYYY-MM-DD'),  3)
and l_returnflag = 'R'
and c_nationkey = n_nationkey
group by
c_custkey,
c_name,
c_acctbal,
c_phone,
n_name,
c_address,
c_comment
order by
revenue desc)
where rownum <= 20

C_CUSTKEY            C_NAME                  REVENUE
C_ACCTBAL            N_NAME
C_ADDRESS                        C_PHONE
C_COMMENT
57040.00             Customer#000057040      734235.25
632.87               JAPAN
Eioyzjf4pp                       22-895-641-3466
requests sleep blithely about the furiously i
143347.00            Customer#000143347      721002.69
2557.47              EGYPT
1aReFYv,Kw4                      14-742-935-3718
fluffily bold excuses haggle finally after the u
60838.00             Customer#000060838      679127.31
2454.77              BRAZIL
64EaJ5vMAHWJlBOxJklpNc2RJiWE           12-913-494-9813
furiously even pinto beans integrate under the ruthless foxes; ironic, even dolphins across the slyl
101998.00            Customer#000101998      637029.57
3790.89              UNITED KINGDOM
01c9CILnNtfOQYmZj                33-593-865-6378
accounts doze blithely! enticing, final deposits sleep blithely special accounts. slyly express
accounts pla
125341.00            Customer#000125341      633508.09
4983.51              GERMANY
S29ODD6bceU8QSuuEJznkNaK              17-582-695-5962
quickly express requests wake quickly blithely
25501.00             Customer#000025501      620269.78
7725.04              ETHIOPIA
  W556MXuoiaYCCZamJI,Rn0B4ACUGdkQ8DZ     15-874-808-6793
quickly special requests sleep evenly among the special deposits. special deposi
115831.00            Customer#000115831      596423.87
5098.10              FRANCE
rFeBbEEyk dl ne7zV5fDrmiq1oK09wV7pxqCgIc 16-715-386-3788
carefully bold excuses sleep alongside of the thinly idle
---------- lines deleted ----------

HgiV0phqhaIa9aydNoIlb                29-915-458-2654
instructions nag quickly. furiously bold accounts cajol


20 rows processed.

Query Processed in 5.52 seconds.


Ended Executing this Stream at Thu Sep  8 14:20:02 2005


Stream Started at 1126207196.72
Stream Ended at 1126207202.24
Stream Processed in 5.52 seconds


SQL statements processed: 1


---------------------------------------------
11.log
---------------------------------------------
Begin Execution at Thu Sep  8 14:20:02 2005


-- 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

---------- lines deleted ----------

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.
Query Processed in 3.07 seconds.


Ended Executing this Stream at Thu Sep  8 14:20:05 2005


Stream Started at 1126207202.30
Stream Ended at 1126207205.37
Stream Processed in 3.07 seconds


SQL statements processed: 1


---------------------------------------------
12.log
---------------------------------------------
Begin Execution at Thu Sep  8 14:20:05 2005



-- using default substitutions


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

L_SHIPMODE HIGH_LINE_COUNT       LOW_LINE_COUNT
MAIL      6202.00               9324.00
SHIP      6200.00               9262.00


2 rows processed.
Query Processed in 4.19 seconds.


Ended Executing this Stream at Thu Sep  8 14:20:09 2005


Stream Started at 1126207205.43
Stream Ended at 1126207209.62
Stream Processed in 4.19 seconds


SQL statements processed: 1


---------------------------------------------
13.log
---------------------------------------------
Begin Execution at Thu Sep  8 14:20:09 2005



-- 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.
Query Processed in 2.91 seconds.


Ended Executing this Stream at Thu Sep  8 14:20:12 2005


Stream Started at 1126207209.67
Stream Ended at 1126207212.58
Stream Processed in 2.91 seconds


SQL statements processed: 1


---------------------------------------------
14.log
---------------------------------------------
Begin Execution at Thu Sep  8 14:20:12 2005



-- using default substitutions


select
        100.00 * sum(case
                when p_type like 'PROMO%'
                        then l_extendedprice * (1 - l_discount)
                else 0
        end) / sum(l_extendedprice * (1 - l_discount)) as promo_revenue
from
        lineitem,
        part
```

```
where
        l_partkey = p_partkey
        and l_shipdate >= date '1995-09-01'
        and l_shipdate < date '1995-09-01' + interval '1' month

PROMO_REVENUE
16.38


1 row processed.
Query Processed in 0.22 seconds.


Ended Executing this Stream at Thu Sep  8 14:20:12 2005


Stream Started at 1126207212.64
Stream Ended at 1126207212.86
Stream Processed in 0.22 seconds


SQL statements processed: 1


---------------------------------------------
15.log
---------------------------------------------
Begin Execution at Thu Sep  8 14:20:12 2005



-- using default substitutions


with   revenue
as (select
l_suppkey supplier_no,
sum(l_extendedprice * (1 - l_discount)) total_revenue
from
lineitem
where
        l_shipdate >= date '1996-01-01'
        and l_shipdate < date '1996-01-01' + interval '3' month
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
Wp34zim9qYFbVctdW                   20-469-856-8873 1772627.21


1 row processed.
Query Processed in 1.54 seconds.


Ended Executing this Stream at Thu Sep  8 14:20:14 2005


Stream Started at 1126207212.91
Stream Ended at 1126207214.46
Stream Processed in 1.54 seconds


SQL statements processed: 1


---------------------------------------------
16.log
---------------------------------------------
Begin Execution at Thu Sep  8 14:20:14 2005



-- 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

---------- lines deleted ----------

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.
Query Processed in 0.99 seconds.


Ended Executing this Stream at Thu Sep  8 14:20:15 2005


Stream Started at 1126207214.69
Stream Ended at 1126207215.68
Stream Processed in 0.99 seconds


SQL statements processed: 1


---------------------------------------------
17.log
---------------------------------------------
Begin Execution at Thu Sep  8 14:20:16 2005



-- 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.
Query Processed in 4.66 seconds.


Ended Executing this Stream at Thu Sep  8 14:20:21 2005


Stream Started at 1126207216.41
Stream Ended at 1126207221.07
Stream Processed in 4.66 seconds


SQL statements processed: 1


---------------------------------------------
18.log
---------------------------------------------
Begin Execution at Thu Sep  8 14:20:21 2005



-- 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

---------- lines deleted ----------

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.
Query Processed in 5.73 seconds.


Ended Executing this Stream at Thu Sep  8 14:20:26 2005


Stream Started at 1126207221.13
Stream Ended at 1126207226.86
Stream Processed in 5.73 seconds


SQL statements processed: 1


---------------------------------------------
19.log
---------------------------------------------
Begin Execution at Thu Sep  8 14:20:26 2005



-- using default substitutions


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

REVENUE
3083843.06


1 row processed.
Query Processed in 4.86 seconds.


Ended Executing this Stream at Thu Sep  8 14:20:31 2005


Stream Started at 1126207226.92
Stream Ended at 1126207231.77
Stream Processed in 4.86 seconds


SQL statements processed: 1


---------------------------------------------
1.log
---------------------------------------------
Begin Execution at Thu Sep  8 14:19:12 2005
```

```
-- using default substitutions


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

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


4 rows processed.
Query Processed in 8.13 seconds.


Ended Executing this Stream at Thu Sep  8 14:19:20 2005


Stream Started at 1126207152.09
Stream Ended at 1126207160.22
Stream Processed in 8.13 seconds


SQL statements processed: 1


-----------------------------------------------
20.log
-----------------------------------------------
Begin Execution at Thu Sep  8 14:20:31 2005



-- using default substitutions


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

S_NAME                    S_ADDRESS
Supplier#000000020        iybAE,RmTymrZVYaFZva2SH,j
Supplier#000000091        YV45D7TkfdQanOOZ7q9QxkyGUapU1oOWU6q3
Supplier#000000197        YC2Acon6kjY3zj3Fbxs2k4Vdf7X0cd2F
Supplier#000000226        83qOdU2EYRdPQAQhEtn GRZEd
Supplier#000000285        Br7e1nnt1yxrw6ImgpJ7YdhFDjuBf
Supplier#000000378        FfbhyCxWvcPrO8ltp9
Supplier#000000402        i9Sw4DoyMhzhKXCH9By,AYSgmD
Supplier#000000530        0qwCMwobKY OcmLyfRXlagA8ukENJv,
Supplier#000000688        D fw5ocppmZpYBBIPI718hCihLDZ5KhKX
Supplier#000000710        f19YPvOyb QoYwjKC,oPycpGfieBAcwKJo
Supplier#000000736        l6i2nMwVuovfKnuVgaSGK2rDy65DlAFLegiL7
Supplier#000000761        zlSLelQUj2XrvTTFnv7WAcYZGvvMTx882d4
Supplier#000000884        bmhEShejaS
Supplier#000000887        urEaTejH5POADP2ARrf
Supplier#000000935        ij98czM 2KzWe7dDTOxB8sq0UfCdvrX
Supplier#000000975        ,AC e,tBpNwKb5xMUzeohxlRn, hdZJo73gFQF8y


---------- lines deleted ----------

Supplier#000009799         4wNjXGa4OKWl
Supplier#000009811        E3iuyq7UnZxU7oPZIe2Gu6
Supplier#000009812        APFRMy3lCbgFga53n5t9DxzFPQPgnjrGt32
Supplier#000009862        rJzweWeN58
Supplier#000009868        ROjGgx5gvtkmnUUoeyy7v
Supplier#000009869        ucLqxzrpBTRMewGSM29t0rNTM30g1Tu3Xgg3mKag
Supplier#000009899        7XdpAHrzr1t,UQFZE
Supplier#000009974        7wJ,J5DKcxSU4Kp1cQLpbcAvB5AsvKT


204 rows processed.
Query Processed in 2.12 seconds.


Ended Executing this Stream at Thu Sep  8 14:20:33 2005


Stream Started at 1126207231.83
Stream Ended at 1126207233.94
Stream Processed in 2.12 seconds


SQL statements processed: 1


-----------------------------------------------
21.log
-----------------------------------------------
Begin Execution at Thu Sep  8 14:20:34 2005



-- 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

---------- lines deleted ----------

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.
Query Processed in 16.98 seconds.


Ended Executing this Stream at Thu Sep  8 14:20:50 2005


Stream Started at 1126207234.01
Stream Ended at 1126207250.98
Stream Processed in 16.98 seconds


SQL statements processed: 1


--------------------------------------------
22.log
--------------------------------------------
Begin Execution at Thu Sep  8 14:20:51 2005



-- using default substitutions


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

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


7 rows processed.
Query Processed in 2.53 seconds.


Ended Executing this Stream at Thu Sep  8 14:20:53 2005


Stream Started at 1126207251.04
Stream Ended at 1126207253.57
Stream Processed in 2.53 seconds


SQL statements processed: 1


--------------------------------------------
2.log
--------------------------------------------
Begin Execution at Thu Sep  8 14:19:20 2005



-- using default substitutions


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

```
and s_nationkey = n_nationkey
and n_regionkey = r_regionkey
and r_name = 'EUROPE'
)
order by
s_acctbal desc,
n_name,
s_name,
p_partkey
)
where rownum <= 100

S_ACCTBAL          S_NAME                N_NAME
P_PARTKEY          P_MFGR
S_ADDRESS                        S_PHONE
S_COMMENT
9938.53            Supplier#000005359    UNITED KINGDOM
185358.00          Manufacturer#4
QKuHYh,vZGiwu2FWEJoLDx04              33-429-790-6131
blithely silent pinto beans are furiously. slyly final deposits acros
9937.84            Supplier#000005969    ROMANIA
108438.00          Manufacturer#1
ANDENSOSmk,miq23Xfb5RWt6dvUcvt6Qa     29-520-692-3537
carefully slow deposits use furiously. slyly ironic platelets above the ironic
9936.22            Supplier#000005250    UNITED KINGDOM
249.00             Manufacturer#4
B3rqp0xbSEim4Mpy2RH J                 33-320-228-2957
blithely special packages are. stealthily express deposits across the closely final instructi
9923.77            Supplier#000002324    GERMANY
29821.00           Manufacturer#4
y3OD9UywSTOk                          17-779-299-1839
quickly express packages breach quiet pinto beans. requ
9871.22            Supplier#000006373    GERMANY
43868.00           Manufacturer#5
J8fcXWsTqM                            17-813-485-8637
never silent deposits integrate furiously blit
9870.78            Supplier#000001286    GERMANY
81285.00           Manufacturer#2
YKA,E2fjiVd7eUrzp2Ef8j1QxGo2DFnosaTEH  17-516-924-4574
final theodolites cajole slyly special,

---------- lines deleted ----------

7871.50            Supplier#000007206    RUSSIA
104695.00          Manufacturer#1
3w fNCnrVmvJjE95sgWZzvW               32-432-452-7731
furiously dogged pinto beans cajole. bold, express notornis until the slyly pending
7852.45            Supplier#000005864    RUSSIA
8363.00            Manufacturer#4
WCNfBPZeSXh3h,c                       32-454-883-3821
blithely regular deposits
7850.66            Supplier#000001518    UNITED KINGDOM
86501.00           Manufacturer#1
ONda3YJiHKJOC                         33-730-383-3892
furiously final accounts wake carefully idle requests. even dolphins wake acc
7843.52            Supplier#000006683    FRANCE
11680.00           Manufacturer#4
2Z0JGkiv01Y00oCFwUGfviIbhzCdy         16-464-517-8943
carefully bold accounts doub


100 rows processed.
Query Processed in 1.48 seconds.


Ended Executing this Stream at Thu Sep  8 14:19:21 2005


Stream Started at 1126207160.28
Stream Ended at 1126207161.75
Stream Processed in 1.48 seconds


SQL statements processed: 1


-----------------------------------------
3.log
-----------------------------------------
Begin Execution at Thu Sep  8 14:19:21 2005



-- using default substitutions


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

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


10 rows processed.
Query Processed in 4.11 seconds.


Ended Executing this Stream at Thu Sep  8 14:19:25 2005


Stream Started at 1126207161.82
Stream Ended at 1126207165.93
Stream Processed in 4.11 seconds


SQL statements processed: 1


-----------------------------------------
4.log
-----------------------------------------
Begin Execution at Thu Sep  8 14:19:25 2005



-- using default substitutions


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

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


5 rows processed.
Query Processed in 5.04 seconds.


Ended Executing this Stream at Thu Sep  8 14:19:31 2005
```

```
Stream Started at 1126207165.99
Stream Ended at 1126207171.03
Stream Processed in 5.04 seconds


SQL statements processed: 1


---------------------------------------------
5.log
---------------------------------------------
Begin Execution at Thu Sep  8 14:19:31 2005



-- using default substitutions


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

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


5 rows processed.
Query Processed in 5.61 seconds.


Ended Executing this Stream at Thu Sep  8 14:19:36 2005


Stream Started at 1126207171.08
Stream Ended at 1126207176.69
Stream Processed in 5.61 seconds


SQL statements processed: 1


---------------------------------------------
6.log
---------------------------------------------
Begin Execution at Thu Sep  8 14:19:36 2005



-- using default substitutions


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

REVENUE
123141078.23


1 row processed.
Query Processed in 0.73 seconds.


Ended Executing this Stream at Thu Sep  8 14:19:37 2005


Stream Started at 1126207176.74
Stream Ended at 1126207177.47
Stream Processed in 0.73 seconds


SQL statements processed: 1


---------------------------------------------
7.log
---------------------------------------------
Begin Execution at Thu Sep  8 14:19:37 2005



-- using default substitutions


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

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


4 rows processed.
Query Processed in 3.87 seconds.


Ended Executing this Stream at Thu Sep  8 14:19:41 2005


Stream Started at 1126207177.52
Stream Ended at 1126207181.39
Stream Processed in 3.87 seconds


SQL statements processed: 1


---------------------------------------------
8.log
---------------------------------------------
```

```
Begin Execution at Thu Sep  8 14:19:41 2005


-- using default substitutions


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

O_YEAR              MKT_SHARE
1995.00             0.03
1996.00             0.04


2 rows processed.
Query Processed in 5.52 seconds.


Ended Executing this Stream at Thu Sep  8 14:19:46 2005


Stream Started at 1126207181.44
Stream Ended at 1126207186.96
Stream Processed in 5.52 seconds


SQL statements processed: 1


--------------------------------------------
9.log
--------------------------------------------
Begin Execution at Thu Sep  8 14:19:47 2005


-- 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
BRAZIL              1998.00             27217924.38
BRAZIL              1997.00             48378669.20
BRAZIL              1996.00             50482870.36
BRAZIL              1995.00             47623383.63
BRAZIL              1994.00             47840165.73
BRAZIL              1993.00             49054694.04
BRAZIL              1992.00             48667639.08


---------- lines deleted ----------

VIETNAM             1995.00             49658284.61
VIETNAM             1994.00             50596057.26
VIETNAM             1993.00             50953919.15
VIETNAM             1992.00             49613838.32


175 rows processed.
Query Processed in 9.64 seconds.


Ended Executing this Stream at Thu Sep  8 14:19:56 2005


Stream Started at 1126207187.02
Stream Ended at 1126207196.66
Stream Processed in 9.64 seconds


SQL statements processed: 1
```

Appendix E: Seed amd Input Parameters

```
---------------------------------------------
qp1.0
---------------------------------------------
14 1995-02-01
2 26 TIN EUROPE
9 green
20 linen 1994-01-01 FRANCE
6 1994-01-01 0.06 25
17 Brand#13 LG BOX
18 314
8 KENYA AFRICA MEDIUM BURNISHED TIN
21 INDIA
13 pending requests
3 HOUSEHOLD 1995-03-19
22 16 20 10 33 19 14
11
16 Brand#51 LARGE PLATED 13 26 18
1 39 9 27 45
4 1996-01-01
11 SAUDI ARABIA 0.0000003333
15 1997-10-01
1 100
10 1994-05-01
19 Brand#23 Brand#22 Brand#34 9 13 22
5 MIDDLE EAST 1994-01-01
7 JORDAN KENYA
12 SHIP AIR 1995-01-01

---------------------------------------------
qp1.1
---------------------------------------------
21 ALGERIA
3 AUTOMOBILE 1995-03-05
18 315
5 AFRICA 1994-01-01
11 INDIA 0.0000003333
7 ETHIOPIA GERMANY
6 1994-01-01 0.03 24
20 tan 1993-01-01 VIETNAM
17 Brand#15 LG PACK
12 FOB RAIL 1995-01-01
16 Brand#31 PROMO POLISHED 16 3 42
6 18 30 19 50
15 1995-07-01
13 pending requests
10 1993-02-01
2 14 STEEL AFRICA
8 ETHIOPIA AFRICA SMALL BRUSHED TIN
14 1995-05-01
19 Brand#25 Brand#15 Brand#23 4 14 29
9 floral
22 31 15 24 19 21 30
32
1 108
4 1993-10-01

---------------------------------------------
qp1.2
---------------------------------------------
6 1995-01-01 0.09 25
17 Brand#12 LG DRUM
14 1995-08-01
16 Brand#21 MEDIUM ANODIZED 19 13 36
38 14 50 45 42
19 Brand#22 Brand#43 Brand#23 9 15 25
10 1993-12-01
9 dark
2 2 BRASS EUROPE
15 1993-04-01
8 RUSSIA EUROPE SMALL PLATED TIN
5 AMERICA 1995-01-01
22 25 18 10 32 23 22
14
12 TRUCK RAIL 1995-01-01
7 RUSSIA ETHIOPIA
13 pending accounts
18 313
1 116
4 1996-04-01
20 ghost 1996-01-01 IRAN
3 HOUSEHOLD 1995-03-21
11 VIETNAM 0.0000003333
21 PERU

---------------------------------------------
qp1.3
---------------------------------------------
8 KENYA AFRICA SMALL ANODIZED TIN
5 ASIA 1995-01-01
4 1994-01-01
6 1995-01-01 0.06 25
17 Brand#14 MED BOX
7 KENYA EGYPT
1 63
18 314
22 10 17 16 34 11 25
18
14 1995-11-01
9 chocolate
10 1994-09-01
15 1995-11-01
11 INDONESIA 0.0000003333
20 red 1995-01-01 ALGERIA
2 40 NICKEL AMERICA
21 IRAN
19 Brand#34 Brand#31 Brand#22 5 17 21
13 pending accounts
16 Brand#51 ECONOMY BURNISHED 4 38 50
23 41 45 22 34
12 RAIL TRUCK 1995-01-01
3 AUTOMOBILE 1995-03-07

---------------------------------------------
qp1.4
---------------------------------------------
5 EUROPE 1995-01-01
21 BRAZIL
14 1996-03-01
19 Brand#31 Brand#14 Brand#11 10 18 29
15 1993-07-01
17 Brand#11 MED PACK
12 AIR RAIL 1996-01-01
6 1995-01-01 0.04 24
4 1996-08-01
9 blush
8 FRANCE EUROPE STANDARD POLISHED NICKEL
16 Brand#31 STANDARD POLISHED 36 14 10
11 7 24 48 44
11 RUSSIA 0.0000003333
2 27 TIN EUROPE
10 1993-06-01
18 312
1 71
13 pending accounts
7 FRANCE BRAZIL
22 20 17 16 33 28 22
18
3 FURNITURE 1995-03-23
20 coral 1993-01-01 MOROCCO

---------------------------------------------
qp1.5
---------------------------------------------
21 ROMANIA
15 1996-02-01
4 1994-05-01
6 1995-01-01 0.09 25
7 UNITED KINGDOM ARGENTINA
16 Brand#21 MEDIUM BRUSHED 30 1 9
5 10 49 24 2
19 Brand#33 Brand#41 Brand#15 5 19 25
18 313
14 1996-06-01
22 14 31 34 18 16 23
19
11 IRAN 0.0000003333
13 pending accounts
3 AUTOMOBILE 1995-03-09
1 79
2 15 COPPER AMERICA
5 MIDDLE EAST 1995-01-01
8 UNITED KINGDOM EUROPE STANDARD BURNISHED
NICKEL
20 moccasin 1996-01-01 EGYPT
12 REG AIR TRUCK 1996-01-01
17 Brand#13 MED DRUM
10 1994-03-01
9 azure
```

```
-------------------------------------------
qp1.6
-------------------------------------------
10          1994-12-01
3           FURNITURE 1995-03-25
15          1993-11-01
13          pending   accounts
6           1996-01-01 0.06       25
8           MOROCCO   AFRICA      PROMO BRUSHED NICKEL
9           wheat
7           MOROCCO   UNITED STATES
4           1996-12-01
11          UNITED KINGDOM       0.0000003333
22          25        11         30          15          32          20
            24
18          315
12          SHIP      TRUCK      1996-01-01
1           87
5           AMERICA   1996-01-01
16          Brand#51  PROMO BURNISHED 16          38          11
            45        50         41          35          47
2           3         BRASS      MIDDLE EAST
14          1996-09-01
19          Brand#41  Brand#34   Brand#55    1           20          21
20          almond    1995-01-01 ROMANIA
17          Brand#15  JUMBO BOX
21          IRAQ

-------------------------------------------
seed
-------------------------------------------
0907150845
```

Appendix F: Benchmark Scripts

```
---------------------------------------------
2asmstart
---------------------------------------------
#!/bin/ksh

. $FRAME_PATH/env
export ORACLE_SID=$ASM_SID
sqlplus /NOLOG << EOF
!date
set timing on
connect / as sysdba
startup pfile=$ORACLE_HOME/dbs/init${ASM_SID}.ora
!date
exit
EOF
cpall $ORACLE_HOME/bin/oracle
for i in $SECONDARY_NODES
do
ssh $i -n $KIT_DIR/rasmstart
done



---------------------------------------------
2shut
---------------------------------------------
#!/bin/ksh
. $FRAME_PATH/env

if [ "$1" = "abort" ]; then
for i in $SECONDARY_NODES
do
ssh $i -n /home/oracle/frame/bin/tshut
done
sqlplus << !
connect / as sysdba
shutdown abort
exit
!
else
for i in $SECONDARY_NODES
do
ssh $i -n /home/oracle/frame/bin/tshut abort
done
sqlplus << !
connect / as sysdba
shutdown immediate
exit
!
fi


---------------------------------------------
2start
---------------------------------------------
#!/bin/ksh

. $FRAME_PATH/env

tstart
echo $SECONDARY_NODES
for i in $SECONDARY_NODES
do
ssh $i -n /home/oracle/frame/bin/tstart
done



---------------------------------------------
qexecpl.c
---------------------------------------------
#ifdef RCSID
static char *RCSid =
   "$Header: qexecpl.c 17-oct-2001.09:29:47 mpoess Exp $ ";
#endif /* RCSID */

/* Copyright (c) 1999, 2001, Oracle Corporation.  All rights reserved.  */

/*

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

  DESCRIPTION
    SQL Execution Engine, Oracle v8,  OCI version

  PRIVATE FUNCTION(S)
    <list of static functions defined in .c file - with one-line descriptions>

  MODIFIED   (MM/DD/YY)
    mpoess    10/17/01 - add serialization level in SQLinit
    mpoess    02/22/01 - add linux changes
    mpoess    08/05/99 - make compile
    mpoess    11/13/98 - fix pddl statement
    pswong    02/19/97 - migrating to version 8
    pswong    04/02/96 - more polishing
    pswong    03/25/96 - polish up
    pswong    03/06/96 - created

*/

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

#include "qexecpl.h"

/* Function Prototypes */

extern double gettime();

/* function prototypes from gen.c */

int get_statement();

/* Declare error handling functions */

void sql_error();

/* Other prototypes */

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

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

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

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

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

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

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

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

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

sltype slist[MAX_SEL_LIST];  /* Array for describing Select List         */
```

```
dltype *dlist[MAX_SEL_LIST]; /* Array of ptrs for Defining Select List   */

char stmt[SQL_LEN];       /* The SQL statement or comment line.  */
char qn[4];               /* Number of the query being executed  */
char qnp[4];              /* Number of the previous query executed  */
char cmnt[5000];          /* Buffer to save the comment.         */
#ifdef LINUX
FILE *qtemp;      /* fd for query template              */
FILE *logfile;   /* log and report files               */
FILE *rep;
#else
FILE *qtemp = stdin;      /* fd for query template              */
FILE *logfile = stdout;   /* log and report files               */
FILE *rep = stdout;
#endif
void *defbuf;             /* Buffer pointer for ODEFIN          */
int deflen = 0;           /* Size of data type for ODEFIN       */
int deftype = 1;          /* Oracle type number for ODEFIN      */

int pfmem = PFMEMSIZE;    /* Memory to prefetch rows            */

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

/* OCI handles */

OCIEnv *tpcenv = NULL;
OCIServer *tpcsrv = NULL;
OCIError *errhp = NULL;
OCISvcCtx *tpcsvc = NULL;
OCISession *tpcusr = NULL;
OCIStmt *curq = NULL;
OCIStmt *cur_dml = NULL;
OCIStmt *cur_ddl = NULL;
OCIParam *tpcpar = NULL;

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

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

void usage() {

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



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

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

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

  /* Rollback just in case */

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

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

  SQLexit();

  exit(1);
}

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

  int i,pos,pos2;
  int retcode;      /* Return code for get_statement */
#ifdef LINUX
  logfile=fopen("/dev/stdout","w");
  qtemp=fopen("/dev/stdin","rw");
  rep=fopen("/dev/stdout","w");
#endif
  /* Initialize some variables */

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

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

  strcpy(logname, argv[1]);

  /* Process  optional parameters */

  argc -= 1;
  argv += 1;

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

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

  SQLinit();

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

  time(&tim);
  fprintf(logfile, "Begin Execution at %s\n\n", ctime(&tim));
  fprintf(rep, "Begin Executing this Stream at %s\n\n", ctime(&tim));
  /* Get the next statement and start processing it */

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

```
  switch (retcode) {

    /* If this is a comment, skips it */
  case COMMENT:
    /*if (end_flag) {
                end_flag = 0;  */    /* reset query end flag */
                /* save the comment so that we can print it out later on  */
                /* strcpy(cmnt, stmt);
                break;
    } */
    if (stmt[3]== '@') {
      pos=4;
      strcpy(qnp,qn);
      while (stmt[pos] != ')') {
        pos++;
      }
      pos2=0;
      pos++;
      while (stmt[pos] != '.') {
        /*printf ("qn %d %c \n",pos2,stmt[pos]);*/
        qn[pos2]=stmt[pos];
        pos2++;
        pos++;
      }
      qn[pos2] = 0;
      /* printf("found a new query: %s\n",qn); */
    }
    /* save the comment so that we can print it out later on  */
    strcat(cmnt, stmt);
    break;

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

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

    /* Executes the query */
    SQLexec();

    stmt_cnt++;
    qry_cnt++;
    fflush(rep);
    fflush(logfile);
    /*
    fprintf(logfile,"\nStatement Started at %.2f\n", s_tr_start);
    fprintf(logfile,"Statement Ended at %.2f\n", s_tr_end);

    fprintf(logfile,"Statement Processed in %.2f seconds.\n",
                    (s_tr_end - s_tr_start));
    fprintf(rep, "Query %s: Execution Time: %.2f started %.2f ended %.2f\n",
              qn,(s_tr_end - s_tr_start)s_tr_start,s_tr_end);
    fflush(rep);
    fflush(logfile);*/
    break;

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

/* Get Timing for the last query */

tr_end = gettime();

fprintf(logfile,"Query Processed in %.2f seconds.\n\n",(tr_end - s_tr_start));

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

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

/* fprintf(rep, "Query %s : Execution time %.2f\n", qn,(tr_end - s_tr_start));*/
fprintf(rep, "Query %s: Execution Time: %.2f started %.2f ended %.2f\n",
              qn,(tr_end - s_tr_start),s_tr_start,tr_end);

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

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

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

  fflush(rep);
  fflush(logfile);

  /* Close the query template file */

  fclose(qtemp);

  /*  Disconnect from ORACLE. */

  SQLexit();
  exit(0);
}


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

void SQLinit() {

  int i;

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

  for (i=0; i<MAX_SEL_LIST; i++) {
    if (i < MAX_PREALLOC) {
      dlist[i] = (dltype *) memalloc (sizeof(dltype));
      dlist[i]->defhdl = NULL;
/*      OCIhalloc(curq,&(dlist[i]->defhdl),OCI_HTYPE_DEFINE); */
    }
    else
      dlist[i] = NULL;
  }

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

  (void) OCIInitialize(OCI_DEFAULT,(dvoid *)0,0,0,0);

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

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

  /* get username and password */

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

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

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

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

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

/*
  if ((status=OCILogon((OCIEnv *)tpcenv,(OCIError *)errhp,(OCISvcCtx *)tpcsvc,
                       (text *)logname, strlen(logname), (text *)passwd,
                       strlen(passwd), (text *) 0, 0)) != OCI_SUCCESS)
    sql_error(errhp, status, 1);
*/
  printf("\nConnected to ORACLE as user: %s\n\n", logname);

}



/* SQLexec()  Executes the SQL statement.                          */
/*           Parse the SQL statement.                          */
```

```
/*          If DDL or DML statements, execute right away.            */
/*          Else describe and define select list outputs,            */
/*               execute and fetch results.                          */

void SQLexec()
{
  int i;
  ub2 stmttyp = OCI_STMT_SELECT;     /* default is a SELECT statement */

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

  if (qry_cnt) {
    time(&tim);
    s_tr_end = gettime();
    fprintf(logfile,"Query Processed in %.2f seconds.\n\n",
     (s_tr_end - s_tr_start));

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

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

    /*fprintf(rep, "Query %s : Execution time %.2f\n", qnp,(s_tr_end - s_tr_start));*/
    fprintf(rep, "Query %s: Execution Time: %.2f started %.2f ended %.2f\n",
                qnp,(s_tr_end - s_tr_start),s_tr_start,s_tr_end);

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

    /* Fix for Q15 */
    fflush(logfile);
    fflush(rep);

    strcpy(qnp,qn);

    fflush(logfile);
    fflush(rep);
  }
  else
    tr_start = gettime();

  s_tr_start = gettime();

  /* prepare the statement */

  if ((status = OCIStmtPrepare(curq, errhp, (text*) stmt, (ub4) strlen(stmt),
                                                  OCI_NTV_SYNTAX, OCI_DEFAULT)) !=
OCI_SUCCESS)
    sql_error(errhp,status,1);

  /* Prints the query text and comment to the logfile */

  fprintf(logfile, "\n%s\n", cmnt);
  cmnt[0]=0;
  fprintf(logfile, "\n%s\n", stmt);

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

  OCIaget(curq,OCI_HTYPE_STMT,&stmttyp,NULL,OCI_ATTR_STMT_TYPE,errhp);

  if (stmttyp != OCI_STMT_SELECT) {
    OCIsexec(tpcsvc,curq,errhp,1);
    return;
  }

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

  /* first let's execute it to get the select-list definition */

  OCIaset(curq, OCI_HTYPE_STMT, &pfmem, 0, OCI_ATTR_PREFETCH_MEMORY,
errhp);

  OCIsexec(tpcsvc,curq,errhp,0);

  num_sel_list = define_output_variables();

  /* Executes the query and fetches the rows */

  (void) process_select_list(num_sel_list);

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

  OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,OCI_ATTR_ROW_COUNT,errhp);

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

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

  /* reset set_fetchrows */

  num_to_fetch = -1;

}


void SQLexit() {

  int i;

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

  /* free all memory */

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

  /* Flush all output */

  fflush(rep);
  fflush(logfile);

}


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

int define_output_variables()
{

  int i;
  int retflag = 0;

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

    slist[i].buflen = MAX_COLNAME_SIZE;

    if (OCIParamGet(curq, OCI_HTYPE_STMT, errhp, (dvoid **) &tpcpar,
                                                    POS(i)) != OCI_SUCCESS)
      break;


    /* dsize and nullok fields of dlist not used */

    OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].dbsize),
                    NULL, OCI_ATTR_DATA_SIZE, errhp);
    OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].dbtype),
                    NULL, OCI_ATTR_DATA_TYPE, errhp);
    OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].buf),
                    &(slist[i].buflen), OCI_ATTR_NAME, errhp);
    OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].precision),
                    NULL, OCI_ATTR_PRECISION, errhp);
    OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].scale),
                    NULL, OCI_ATTR_SCALE, errhp);

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

/*
    if (slist[i].buflen < MAX_COLNAME_SIZE)
      (slist[i].buf)[slist[i].buflen] = '\0';
*/
    /* Well, we need to allocate for entries for dlist */

    if (i >= MAX_PREALLOC) {
      dlist[i] = (dltype *) memalloc(sizeof(dltype));
      dlist[i]->defhdl = NULL;
    }

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

    switch (slist[i].dbtype) {

    case OCI_TYPECODE_NUMBER:

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

#ifdef HAVE_SCALE
```

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

    break;

  default:

    /* default is character string */

    defbuf = (char **) dlist[i]->sbuf;
    deflen = MAX_STR_LEN;
    deftype = SQLT_STR;
/*      deftype = OCI_TYPECODE_CHAR; */
    break;
  }

  /* Define the column */

  if ((status=OCIDefineByPos(curq,&(dlist[i]->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 %s processed.\n", rows_ret,
                rows_ret == 1 ? "row" : "rows");

}


int get_statement()
{

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

 /* Reset statement buffer */

 stmt[0] = '\0';

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

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

  /* remove blanks */

  str = line;

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

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

  strcat(stmt, str);

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

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

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


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

void *memalloc(size)
    int size;
{

 void *tmp;

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

```
    return tmp;
  }
}



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

  int i, diff;
  char colname[MAX_COLNAME_SIZE];
  int len = 0;          /* Running column length */
  int cwid = 0;

  fprintf(logfile, "\n");

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

    /* extract the column name */

    strncpy((char *)colname, (char *)slist[i].buf, slist[i].buflen);
    colname[slist[i].buflen] = '\0';

    /* format the output a little */

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

    /* do a little bit of formatting */

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

  fprintf(logfile,"\n");
}


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

  int i,j;
  int len;
  int diff;
  int cwid;

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

    len = 0;

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

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

      /* do a little bit of formatting */

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

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


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

void remove_newline(str)
    char *str;
{

  char *p;

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


----------------------------------------
qexecpl.h
----------------------------------------
/*
 * $Header: qexecpl.h 13-nov-2001.17:52:35 mpoess Exp $
 */

/* Copyright (c) 1999, 2001, Oracle Corporation.  All rights reserved.  */

/* NOTE:  See 'header_template.doc' in the 'doc' dve under the 'forms'
      directory for the header file template that includes instructions.
*/

/*
   NAME
     qexecpl.h

   DESCRIPTION
     SQL statement execution front-end header file.

   PUBLIC FUNCTION(S)
     <list of external functions declared/defined - with one-line descriptions>

   PRIVATE FUNCTION(S)
     <list of static functions defined in .c file - with one-line descriptions>

   EXAMPLES

   NOTES
     <other useful comments, qualifications, etc.>

   MODIFIED   (MM/DD/YY)
   mpoess      11/13/01 - change DOP to 84 for DML and DDL
   mpoess      02/22/01 - add linux changes
   mpoess      08/05/99 - make compile
   mpoess      07/15/99 - Creation
   mpoess      07/15/99 - Creation

*/

/*
# ifndef S_ORACLE
#  include <s.h>
# endif
*/
#ifndef QSTREAMPL_H

#define QSTREAMPL_H

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

#include <oratypes.h>

#include <oratypes.h>

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

#ifndef OCI_ORACLE
#include <oci.h>
#endif /* OCI_ORACLE */
/*
#ifdef __STDC__
#include <ociapr.h>
```

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

/* some basic definitions */

#define UNAME_LEN 64
#define MAX_FILE_PATH_LEN 128

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

#ifndef FALSE
#define FALSE 1
#endif /* FALSE */
#ifndef LINUX
#define MAX(x,y) ((x >= y) ? x : y)
#define MIN(x,y) ((x <= y) ? x : y)
#endif
/* defines and typedefs for parsing */

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

/* defines and typedefs for query description */

#define MAX_COLNAME_SIZE 32   /* Maximum length of Column name  */
#define MAX_SEL_LIST 16       /* Maximum items on a select list */

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

/* types for describe */

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

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

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

typedef struct des
{
  ub2 dbsize;
  ub4 buflen;
/*  sb2 dsize;  */
  sb4 scale;
/*  sb2 nullok; */
  OCITypeCode dbtype;
/*  text buf[MAX_COLNAME_SIZE]; */
  text *buf;
  ub1 precision;
} sltype;


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

#define MAX_ARRAY 50       /* Maximum array size for array fetch */
#define PFMEMSIZE 65536    /* Memory size of prefetch buffer     */

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

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

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

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

extern int errno;

#define SQL_LEN 2048

#ifndef NULL
#define NULL 0
#endif

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

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#ifndef ub1
#define ub1 unsigned char
#endif

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

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

/* For get_statement */

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


#define OCIhalloc(envh,hndl,htyp) \
   if((status=OCIHandleAlloc((dvoid *)envh,(dvoid **)hndl,htyp,0,(dvoid
**))!=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 */




---------------------------------------------
rasmstart
---------------------------------------------
export ORACLE_SID=$ASM_SID
sqlplus /NOLOG <<EOF
!date
set timing on
connect / as sysdba
startup pfile=$ORACLE_HOME/dbs/init${ASM_SID}.ora
!date
exit
```

```
EOF


---------------------------------------------
runTPCHall.load
---------------------------------------------
#!/bin/ksh
. $KIT_DIR/env

ECHO=echo

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

RUN_ID_FILE=${KIT_DIR}/audit/r_id

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

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

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

SCRIPT_LOG_FILE=${OUT_DIR}/main.out
RDB_TABLES=${OUT_DIR}/rdbtablest
FIRST_TEN=${OUT_DIR}/firstten
LD1DBCRE=${OUT_DIR}/Ld1dbcre
LD2SCTSO=${OUT_DIR}/Ld2sctso
LD3DAPOP=${OUT_DIR}/Ld3dapop


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

dbcre_10gR2.sh >> $LD1DBCRE
tscre_10gR2.sh >>  $LD2SCTSO
tshut
asmshut
2asmstart
2start
STIME=`$GTIME`
echo "Start: timed load portion `date`" >> $SCRIPT_LOG_FILE
dapop_10gR2.sh >> $LD3DAPOP
$KIT_DIR/audit/gen_seed.sh $KIT_DIR/audit/seed
echo Generated seed: `cat $KIT_DIR/audit/seed` >> $SCRIPT_LOG_FILE
2shut
2asmshut
echo "End: timed load portion `date`" >> $SCRIPT_LOG_FILE
exit

2start  >> $SCRIPT_LOG_FILE
ckpnt.sh

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}

2shut  >> $SCRIPT_LOG_FILE
2asmshut  >> $SCRIPT_LOG_FILE
exit
2asmstart >> $SCRIPT_LOG_FILE
2start  >> $SCRIPT_LOG_FILE
ckpnt.sh
ckpnt.sh


runTPCHpt ${SCALE_FACTOR} 2 ${RUN_ID}

2shut >> $SCRIPT_LOG_FILE
2asmshut >> $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
```

```
---------------------------------------------
runTPCHall.run1
---------------------------------------------
#!/bin/ksh
. $KIT_DIR/env

ECHO=echo

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

RUN_ID_FILE=${KIT_DIR}/audit/r_id

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

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

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

SCRIPT_LOG_FILE=${OUT_DIR}/main.out
RDB_TABLES=${OUT_DIR}/rdbtablest
FIRST_TEN=${OUT_DIR}/firstten
LD1DBCRE=${OUT_DIR}/Ld1dbcre
LD2SCTSO=${OUT_DIR}/Ld2sctso
LD3DAPOP=${OUT_DIR}/Ld3dapop


#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
#
#dbcre_10gR2.sh >> $LD1DBCRE
#tscre_10gR2.sh >>  $LD2SCTSO
#tshut
#asmshut
#2asmstart
#2start
#STIME=`$GTIME`
#echo "Start: timed load portion `date`" >> $SCRIPT_LOG_FILE
#dapop_10gR2.sh >> $LD3DAPOP
#$KIT_DIR/audit/gen_seed.sh $KIT_DIR/audit/seed
#echo Generated seed: `cat $KIT_DIR/audit/seed` >> $SCRIPT_LOG_FILE
#2shut
#2asmshut
#echo "End: timed load portion `date`" >> $SCRIPT_LOG_FILE
#exit

2start  >> $SCRIPT_LOG_FILE
ckpnt.sh

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}

2shut  >> $SCRIPT_LOG_FILE
2asmshut  >> $SCRIPT_LOG_FILE
exit
2asmstart >> $SCRIPT_LOG_FILE
2start  >> $SCRIPT_LOG_FILE
ckpnt.sh
ckpnt.sh


runTPCHpt ${SCALE_FACTOR} 2 ${RUN_ID}

2shut >> $SCRIPT_LOG_FILE
2asmshut >> $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
```

```
---------------------------------------------
runTPCHall.run2
---------------------------------------------
#!/bin/ksh
. $KIT_DIR/env

ECHO=echo

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

RUN_ID_FILE=${KIT_DIR}/audit/r_id

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

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

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

SCRIPT_LOG_FILE=${OUT_DIR}/main.out
RDB_TABLES=${OUT_DIR}/rdbtablest
FIRST_TEN=${OUT_DIR}/firstten
LD1DBCRE=${OUT_DIR}/Ld1dbcre
LD2SCTSO=${OUT_DIR}/Ld2sctso
LD3DAPOP=${OUT_DIR}/Ld3dapop


#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
#
#dbcre_10gR2.sh >> $LD1DBCRE
#tscre_10gR2.sh >>  $LD2SCTSO
#tshut
#asmshut
#2asmstart
#2start
#STIME=`$GTIME`
#echo "Start: timed load portion `date`" >> $SCRIPT_LOG_FILE
#dapop_10gR2.sh >> $LD3DAPOP
#$KIT_DIR/audit/gen_seed.sh $KIT_DIR/audit/seed
#echo Generated seed: `cat $KIT_DIR/audit/seed` >> $SCRIPT_LOG_FILE
#2shut
#2asmshut
#echo "End: timed load portion `date`" >> $SCRIPT_LOG_FILE
#exit

#2start  >> $SCRIPT_LOG_FILE
#ckpnt.sh
#
#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}
#
#2shut  >> $SCRIPT_LOG_FILE
#2asmshut  >> $SCRIPT_LOG_FILE
#exit
2asmstart >> $SCRIPT_LOG_FILE
2start  >> $SCRIPT_LOG_FILE
ckpnt.sh
ckpnt.sh


runTPCHpt ${SCALE_FACTOR} 2 ${RUN_ID}

2shut >> $SCRIPT_LOG_FILE
2asmshut >> $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
```

```
---------------------------------------------
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
---------------------------------------------
#
# $Header: runuf1.sh 25-oct-2001.15:56:04 mpoess Exp $
#
```

```
# runuf1.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation.  All rights reserved.
#
#   NAME
#     runuf1.sh - <one-line expansion of the name>
#
# DESCRIPTION
#   runuf1.sh -l [<path name for reports>] -u [<uid/passwd>]
#             -p [<program>] <run_id> <scale factor> <pair number>
#             <parallelism>
# USAGE
#   To execute UF1.
#
#   NOTES
#     <other useful comments, qualifications, etc.>
#
#   MODIFIED   (MM/DD/YY)
#   mpoess      10/25/01 - change default directory for update sets
#   mpoess      10/17/01 - add support for external tables
#   mpoess      08/15/99 - Creation
#   mpoess      08/15/99 - Creation
#
#
. $KIT_DIR/env
O=${ORACLE_HOME}
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
UTILS_DIR=${KIT_DIR}/utils
LOG_DIR=${UPDATE_DIR}/log
GTIME=${UTILS_DIR}/gtime
SF=${SCALE_FACTOR}
PAR_HINT=16

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 '/home/oracle/dev/ff_update_1/';

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

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


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

insert into orders(
select
o_orderdate         ,
o_orderkey          ,
o_custkey           ,
o_orderpriority     ,
o_shippriority      ,
o_clerk             ,
o_orderstatus       ,
o_totalprice        ,
o_comment
from temp_o_et);

insert into lineitem(
 select
   l_shipdate          ,
   l_orderkey          ,
   l_discount          ,
   l_extendedprice     ,
   l_suppkey           ,
   l_quantity          ,
   l_returnflag        ,
   l_partkey           ,
   l_linestatus        ,
   l_tax               ,
   l_commitdate        ,
   l_receiptdate       ,
   l_shipmode          ,
   l_linenumber        ,
   l_shipinstruct      ,
   l_comment
from temp_l_et);

commit;
drop table temp_l_et;
drop table temp_o_et;

exit;
!

END=`$GTIME`

# Done

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


---------------------------------------------
runuf2.sh
---------------------------------------------
#!/bin/ksh
#
# $Header: runuf2.sh 25-oct-2001.15:56:05 mpoess Exp $
#
# runuf2.sh
```

```
#
# Copyright (c) 1999, 2001, Oracle Corporation.  All rights reserved.
#
#   NAME
#     runuf2.sh - <one-line expansion of the name>
#
# DESCRIPTION
#   runuf2.sh [-u <uid/passwd to login>] [-p <program>] <run_id>
#             <scale factor> <pair number> <parallelism>
# USAGE
#   To execute UF2.
#
#   NOTES
#     <other useful comments, qualifications, etc.>
#
#   MODIFIED   (MM/DD/YY)
#   mpoess      10/25/01 - change default directory for update sets
#   mpoess      10/17/01 - add support for external tables
#   mpoess      08/15/99 - Creation
#   mpoess      08/15/99 - Creation
#
. $KIT_DIR/env
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
UTILS_DIR=${KIT_DIR}/utils
GTIME=${UTILS_DIR}/gtime
LOG_DIR=${UPDATE_DIR}/log
PAR_HINT=16
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 '/home/oracle/dev/ff_update_1/';

drop table temp_okey_et;
drop table temp_okey;

create table temp_okey_et(
   t_orderkey          number
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
            records delimited by newline
            nobadfile
            nologfile
            fields terminated by '|'
            missing field values are null
      )
      location (
      'delete.${SETNUM}'))
reject limit unlimited;

alter table temp_okey_et parallel ${PAR_HINT};


create table temp_okey parallel ${PAR_HINT} nologging as select * from temp_okey_et;

create unique index i_temp_okey on temp_okey (t_orderkey) parallel ${PAR_HINT}  nologging
compute statistics;

analyze table temp_okey estimate statistics sample 1 percent;

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

delete from (select /*+ use_nl(o) */ o.rowid from orders o, temp_okey t where o.o_orderkey =
t.t_orderkey order by 1);

delete from (select /*+ use_nl(l) */ l.rowid from lineitem l,temp_okey t where l.l_orderkey =
t.t_orderkey order by 1);

commit;
drop table temp_okey;
drop table temp_okey_et;
exit;
!

END=`$GTIME`

# Done

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


---------------------------------------------
tshut
---------------------------------------------
#!/bin/ksh
#
# $Header: tshut.sh 08-aug-99.18:06:22 mpoess Exp $
#
# tshut.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#   NAME
#     tshut.sh
#
#   DESCRIPTION
#     shuts down a database
#
#   NOTES
#     <other useful comments, qualifications, etc.>
#
#   MODIFIED   (MM/DD/YY)
#   mpoess      08/08/99 - Creation
#   mpoess      08/08/99 - Creation
#
#!/bin/ksh

if [ "$1" = "abort" ]; then
sqlplus /NOLOG<< !
connect / as sysdba
shutdown abort
exit
!
else
sqlplus /NOLOG<< !
connect / as sysdba
shutdown
exit
!
fi



---------------------------------------------
tstart
---------------------------------------------
#!/bin/ksh
#
# $Header: tstart.sh 08-aug-99.18:05:50 mpoess Exp $
#
# tstart.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#   NAME
#     tstart.sh
#
#   DESCRIPTION
#     starts a database with a specific init.ora or uses the default.
#
#   NOTES
#     <other useful comments, qualifications, etc.>
#
#   MODIFIED   (MM/DD/YY)
#   mpoess      08/08/99 - Creation
#   mpoess      08/08/99 - Creation
#
#!/bin/ksh

set -x

DIR=`pwd`
cd $ORACLE_HOME/dbs

if [ "$1" != "" ]; then
  PFILE="pfile=$ORACLE_HOME/dbs/$1.ora"
else
  PFILE="pfile=$ORACLE_HOME/dbs/init_$ORACLE_SID.ora"
```

```
fi



sqlplus /NOLOG << !
connect / as sysdba
shutdown immediate
startup $PFILE
exit
!

cd $DIR


---------------------------------------------
backup.sh
---------------------------------------------
date
a=1
c=1
for i in 1 2 3 4 5 6 7 8
do
for j in 1 2 3 4
do
   ssh blh$i dd if=/home/oracle/dev/raw/lo_$a of=/backup_$a/lo_$a bs=1024k count=20481 &
   ssh blh$i dd if=/home/oracle/dev/raw/t_$a of=/backup_$a/t_$a bs=1024k count=10241 &
let a+=1
done
b=1
for k in 1a 1b 2a 2b
do


   ssh blh$i dd if=/home/oracle/dev/raw/log_${i}_${k} of=/backup_$b/log_${i}_${k} bs=1024k
count=4097 &let b+=1
done
   ssh blh$i dd if=/home/oracle/dev/raw/undo$i of=/backup_${c}/undo$i bs=1024k count=8197
&
let c+=4
done

d=1
e=1
for i in control1 control2 default ocr quorum sp_0 sys sysaux
do
   ssh blh$d dd if=/home/oracle/dev/raw/$i of=/backup_${e}/$i bs=1024k count=600 &
let d+=1
let e+=4
done
wait
date
```

Appendix G: Price Quotes

Support options and pricing

Server solutions

Red Hat Enterprise Linux server solution subscriptions are available on a per-system annual basis. There are three subscription editions: **Basic, Standard, and Premium**—each with varying support levels and delivery options.

Contact Sales

For **more information** or to order **five or more** units of any version of Red Hat Enterprise Linux please contact Red Hat Sales or call 1-866-273-3428 x45555

Subscription pricing

	Basic Editions	Standard Editions	Premium Editions
Red Hat Enterprise Linux AS			
Platforms: Intel x86, Intel Itanium, Intel EM64T, AMD64, IBM POWER series		**$1499** Buy Now	**$2499** Buy Now
Red Hat Enterprise Linux ES			
Platforms: Intel x86, Intel Itanium, Intel EM64T, and AMD64	**$349** Buy Now	**$799** Buy Now	

Support and features

	Basic Editions	Standard Editions	Premium Editions
Easy ISOs: OS, source, and documentation ISO images	yes	yes	yes
Red Hat Network Update Module Service 1 year	yes	yes	yes
Product updates	yes	yes	yes
Installation and documentation media (CDs)	Optional	yes	yes
Web support	1 year Installation & Basic configuration	24x7	24x7
Phone support	30 days Installation and Basic configuration North Am: 9-9 ET M-F Global: 9-5 GMT/CET M-F	North Am: 9-9 ET M-F Global: 9-5 GMT/CET M-F	24x7 (severity 1)
Web response time/SLA	2 business days	2 business days	1 business day

Phone response time/SLA	1 business day	4 hours	1 hour (severity 1)
Scope of coverage	30 days telephone / 1-year web Installation and Basic configuration	1 year: Standard coverage	1 year: Premium coverage

- Get more information about Enterprise Linux on mainframes

Othayoth, Raghunath

Subject: FW: Please use this one: Oracle Pricing

From: MaryBeth Pierantoni [mailto:mary.beth.pierantoni@oracle.com]
Sent: Friday, September 09, 2005 5:23 PM
To: Othayoth, Raghunath
Subject: Please use this one: Oracle Pricing

Hi Raghu,

To follow is the pricing for the 8N, 1 CPU each (8 CPU total - single core) benchmark with RAC and Partitioning.

Product	Price	Quantity	Extended Price
Oracle Database 10g Enterprise Edition Release 2, Named User Plus for 3 years	10,000	8	80,000
Oracle Real Application Clusters, Named User Plus for 3 years	5,000	8	40,000
Partitioning, Named User Plus for 3 years	2,500	8	20,000
Oracle Database Server Support Package for 3 years	16,000	3	48,000
Oracle Mandatory E-Business Discount			<28,200>
TOTAL			159,800

Oracle Pricing Contact: MaryBeth Pierantoni, mary.beth.pierantoni@oracle.com, 916-315-5081