



Hewlett-Packard Company

TPC BenchmarkTM H Full Disclosure Report

HP BladeSystem 64P
using
Oracle Database 10g Enterprise Edition with
Real Application Cluster and
Partitioning; and
Red Hat Enterprise Linux 4 AS

**First Edition
June 2006**

First Edition – June 2006

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.

© Copyright 2006 Hewlett-Packard Development Company, L.P.

All rights reserved. Permission is hereby granted to reproduce this document in whole or in part provided the copyright notice printed above is set forth in full text or on the title page of each item reproduced.

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 QpH 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 64P DC using 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 @ 3000GB	QthH @ 3000GB	QphH @ 3000GB	\$/QphH @ 3000GB
HP BladeSystem 64P DC	Oracle Database 10g Enterprise Edition R2 with Real Application Cluster and Partitioning, and Red Hat Enterprise Linux 4 AS	\$4,179,238	116,379.3	105,063.0	110,576.5	\$37.80

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 64P DC can 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.



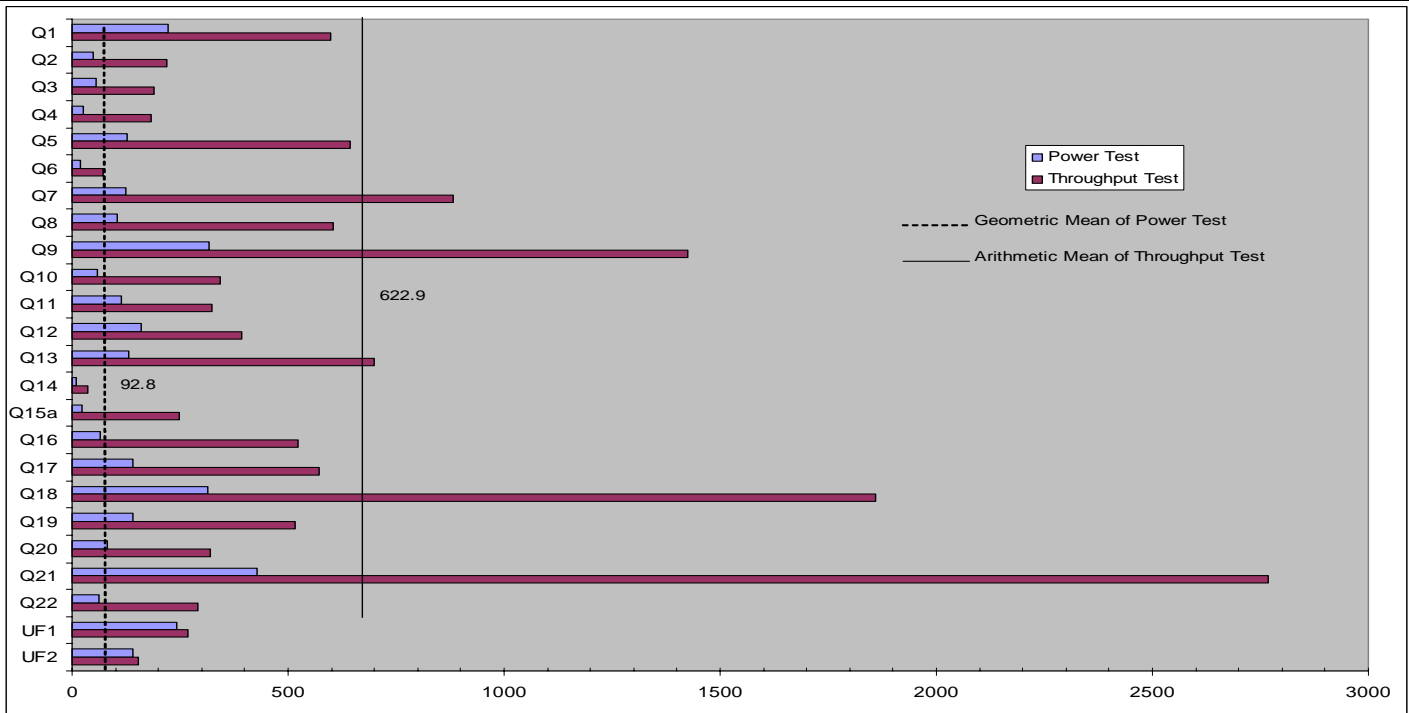
HP BladeSystem 64P

TPC-H Rev. 2.3.0

Report Date:

June 8, 2006

Total System Cost		Composite Query per Hour Metric		Price / Performance	
\$4,179,238 USD		110,576.5 QphH@3000GB		\$37.80 USD \$ / QphH@3000GB	
Database Size	Database Manager	Operating System	Other Software	Availability Date	
3000GB	Oracle Database 10g Release 2, Enterprise Edition with Oracle Real Application Clusters and Partitioning	Red Hat Enterprise Linux 4		June 8, 2006	



Database Load Time = 05:51:14	Load Included Backup: N	Total Data Storage / Database Size = 23.21
RAID (Base tables only): N	RAID (Base tables and auxiliary data structures): N	RAID (All): Y

64 ProLiant server blade BL25p :

Processors (per blade):	1 x Dual-Core AMD Opteron™ processor Model 280 (2.6GHz/1MB L2 cache per core for 2MB total/1 GHz HyperTransport)
Cores (per blade) :	2
Threads (per blade) :	2
Memory (per blade) :	12 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 4/256 SAN Director Switch 129 x hp StorageWorks MSA1000 1806 x 36GB 15krpm HDD Ultra320
Total Storage :	69624 GB



HP BladeSystem 64P

TPC-H Rev. 2.3.0

Report Date:

8-Jun-06

Description	Part Number	Brand	Pricing	Unit Price	Qty	Extended Price	3 yr. Maint. Price
Server Hardware							
HP ProLiant BL25p CTO Blade	382731-405		1	1,493.00	64	95,552	
HP BL25p O285 2.6GHZ DC Kit	406426-B21		1	1,449.00	64	92,736	
HP 8GB REG PC2700 2X4GB Memory	395409-B21		1	7,299.00	64	467,136	
HP 4GB REG PC2700 2 x 2GB Memory	371049-B21		1	1,499.00	64	95,936	
36GB 15Krpm U320 UNI HDD	286776-B22		1	269	128	34,432	
HP BL25/45p Fiber Channel Adapter	381881-B21		1	599	64	38,336	
HP 3y 4h 24x7 ProLiant Svr Blade HW Supp	UD188E		1	295	64		18,880
HP 5642 Unassembled Rack	358254-B21		1	689	16	11,024	
HP BladeSystem p-Class Server Blade Enclosure	243564-B22		1	1,499	8	11,992	
HP BLp-Class Pair RJ-45 Patch Panel	230766-B21		1	1,172.00	8	9,376	
HP BladeSystem Enclosure/Pwr Bundle	390738-B21		1	3,099.00	8	24,792	
HP 3y 4h 24x7 BladeSystem Encl HW Supp	HC032E		1	737	8		5,896
HP T2200 XR High Voltage US UPS	204451-002		1	749	8	5,992	
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	
HP ProCurve Switch 5308xl-48G, 3 16 port copper Gigabit modules (3+2 spares)	J8167A#ABA		1a	7,499	5	37,495	Included
HP ProCurve Switch xl 16 10/100/1000 modules (4+2 spares)	J4907A		1a	2,499	6	14,994	Included
					Subtotal	940,058	24,776
Storage							
HP Storaeworks Modular SAN Array 1000 (129+13 spares)	201723-B22		1	6,995	142	993,290	Included
36GB, 15krpm HDD Ultra320 HP (1806 + 181 spares)	286776-B22		1	269	1987	534,395	Included
5m SW LC/LC FC cable kit	221692-B22		1	82	128	10,496	
15m SW LC/LC FC cable kit	221692-B23		1	103	128	13,184	
2Gb SFF-SW Tmncvr Kit ALL 2GB Small Form Pluggable Adapter Kit	221470-B21		1	199	128	25,472	
HP 3y 4h 24x7 StorWrks MSA 1000 HWSupp	U6357E		1	3,222	0		0
HP 4/256 SAN Director Switch	A7988A		1a	62,999	2	125,998	
HP 4/256 SAN Director 32 port 4G Blade	A7991A		1a	50,999	16	815,984	
HP CPe 3Y 4H 24x7 HW Director2	341444-001		1a	29,164	2		58,328
					Subtotal	2,518,819	58,328
Software							
Oracle Database 10g Enterprise Edition Relase 2, Named User Plus for 3 Years	run-time	Oracle	2	10,000	64 **	640,000	
Oracle Real Application Clusters, Named User Plus for 3 Years	run-time	Oracle	2	5,000	64 **	320,000	
Partitioning, Named User Plus for 3 Years	run-time	Oracle	2	2,500	64 **	160,000	
Database Server Support Package for 3 Years	run-time	Oracle	2	128,000	3		384,000
Red Hat Ent Linux 4 AS Prm 24X7	384956-B21		1	699	192	134,208	Included
					Subtotal	1,254,208	384,000
HP Large Purchase and Net30 discount			1			(610,825)	(14,128)
Oracle Mandatory E-Business Discount			2			(376,000)	
					Total	3,726,261	452,976

Pricing: 1-HP Direct at 17% discount 800-203-7648, 1a-HP Direct at 17% discount. 800-203-7648

Pricing: 2-Oracle at 15% discount, Oracle Pricing Contact: MaryBeth Pierantoni, mary.beth.pierantoni@oracle.com, 916-315-5081

** 64 = 128 * 0.50. Explanation: For the purposes of counting the number of processors which require licensing, an AMD multicore chip with "n" cores shall be determined by multiplying "n" cores by a factor of 0.50.

All discounts are based on US list prices and for similar quantities and configurations.

Oracle Price quote is in Appendix G

3-Year Cost of Ownership in USD: \$4,179,238 USD

QpH Rating: 110,576.5

USD\$ / QpH@3000GB: \$37.80 USD



HP BladeSystem 64P

TPC-H Rev. 2.3.0

Report Date:

June 8, 2006

Numerical Quantities

Measurement Results:

Database Scale Factor = 3000
 Total Data Storage / Database Size = 23.21
 Start of Database Load = 05/30/2006 16:26:23
 End of Database Load = 05/30/2006 22:17:37
 Database Load Time = 05:51:14

Query Streams for Throughput Test = 8
 TPC-H Power = 116,379.3
 TPC-H Throughput = 105,063.0
 TPC-H Composite Query-per-Hour Metric (QphH@3000GB) = 110,576.5
 Total System Price Over 3 Years = \$4,179,238 USD
 TPC-H Price/ Performance Metric (\$/QphH@3000GB) = \$37.80 USD

Measurement Intervals:

Measurement Interval in Throughput Test (Ts) = 18092.0 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	530221737	05/31/2006 7:43:31 05/31/2006 8:29:45	05/31/2006 7:39:27 05/31/2006 7:43:31	05/31/2006 8:29:45 05/31/2006 8:32:05	0:52:38
Stream 1	530221738	05/31/2006 8:32:10 05/31/2006 12:20:49	05/31/2006 12:37:41 05/31/2006 12:42:38	05/31/2006 12:42:38 05/31/2006 12:45:21	4:13:11
Stream 2	530221739	05/31/2006 8:32:10 05/31/2006 12:14:12	05/31/2006 12:45:21 05/31/2006 12:49:49	05/31/2006 12:49:49 05/31/2006 12:52:23	4:20:13
Stream 3	530221740	05/31/2006 8:32:10 05/31/2006 11:54:56	05/31/2006 12:52:23 05/31/2006 12:56:54	05/31/2006 12:56:54 05/31/2006 12:59:33	4:27:23
Stream 4	530221741	05/31/2006 8:32:11 05/31/2006 11:54:52	05/31/2006 12:59:33 05/31/2006 13:04:02	05/31/2006 13:04:02 05/31/2006 13:06:53	4:34:42
Stream 5	530221742	05/31/2006 8:32:10 05/31/2006 12:37:41	05/31/2006 13:06:53 05/31/2006 13:11:31	05/31/2006 13:11:31 05/31/2006 13:14:08	4:41:58
Stream 6	530221743	05/31/2006 8:32:11 05/31/2006 12:34:12	05/31/2006 13:14:08 05/31/2006 13:18:20	05/31/2006 13:18:20 05/31/2006 13:20:38	4:48:27
Stream 7	530221744	05/31/2006 8:32:11 05/31/2006 12:34:49	05/31/2006 13:20:38 05/31/2006 13:24:50	05/31/2006 13:24:50 05/31/2006 13:27:02	4:54:51
Stream 8	530221745	05/31/2006 8:32:10 05/31/2006 12:33:11	05/31/2006 13:27:02 05/31/2006 13:31:17	05/31/2006 13:31:17 05/31/2006 13:33:42	5:01:32



HP BladeSystem 64P

TPC-H Rev. 2.3.0

Report Date:

June 8, 2006

TPC-H Timing Intervals (in seconds)

Query	Stream 0	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7	Stream 8	Min Qi	Max Qi	Avg Qi
Q1	221.4	369.5	398.8	362.9	506.2	463.8	896.5	1056.1	743	362.9	1056.1	599.6
Q2	48.7	357.1	367.4	143.5	217.9	146.9	140.2	153.1	236.1	140.2	367.4	220.3
Q3	55.7	205.4	184.3	216	177.8	84.3	274.8	150.7	229.3	84.3	274.8	190.3
Q4	26.1	64.3	295.1	200.6	204.6	153.5	189.9	198.5	160.7	64.3	295.1	183.4
Q5	127.3	865.2	895.4	688	580.8	407.8	409.9	822.1	471.6	407.8	895.4	642.6
Q6	18.7	51	27.6	68.2	156	104.5	69.6	39.1	54.4	27.6	156.0	71.3
Q7	124.5	994.4	591.4	510.7	627.9	1354.9	961.6	986.2	1028	510.7	1354.9	881.9
Q8	106	696.1	651.8	299.9	586.1	418.1	527.7	1024.9	644.2	299.9	1024.9	606.1
Q9	316	881.9	2068.4	878.5	846	435.9	2668.9	2681	938.6	435.9	2681.0	1424.9
Q10	59	253.6	1169.2	332.9	275.7	96.3	208	144.6	263.6	96.3	1169.2	343.0
Q11	113.1	475.1	283.3	267.1	268.2	409.1	272.5	379.6	224	224.0	475.1	322.4
Q12	161.6	432.8	241.3	373.8	375.8	174.7	698.3	242.2	595.4	174.7	698.3	391.8
Q13	130.5	1089.4	408.8	543.5	993.4	1091.2	373.7	369.3	713.3	369.3	1091.2	697.8
Q14	10.7	33.5	56.5	21.7	35.1	44.1	31.5	24.3	34.1	21.7	56.5	35.1
Q15a	23.8	237.5	112.6	125.5	380.8	511.1	331.9	119.3	170.9	112.6	511.1	248.7
Q16	66.4	453	1127.2	301.7	631.7	435.8	579.1	250.4	402	250.4	1127.2	522.6
Q17	139.2	312.2	461.9	1124.3	647.8	216.6	808.2	292	705.2	216.6	1124.3	571.0
Q18	312.1	1424.8	1468.4	1318.4	1681.5	3617.2	1185.8	1559.2	2609.5	1185.8	3617.2	1858.1
Q19	141.8	744.6	790.6	380.4	399	734.2	349.5	398	321.6	321.6	790.6	514.7
Q20	82.1	229.7	223.3	197.8	351	142.9	384	272.5	747.9	142.9	747.9	318.6
Q21	429.3	3280.7	1239.7	3602.9	1963.9	3108.4	2931.5	3151.6	2877.6	1239.7	3602.9	2769.5
Q22	60.5	266.7	258.9	207.2	254.7	579.1	228.7	243.7	289.6	207.2	579.1	291.1
UF1	243.3	297	267.6	271.4	268.4	277.7	251.6	252.3	254.7	251.6	297.0	267.6
UF2	139.5	163.2	154.1	159.2	171.5	157.4	137.5	131.9	145.8	131.9	171.5	152.6

Table Of Contents

ABSTRACT	II
OVERVIEW	II
STANDARD AND EXECUTIVE SUMMARY STATEMENTS	II
AUDITOR	II
TABLE OF CONTENTS	VII
1.0 GENERAL ITEMS	9
1.1 TEST SPONSOR	9
1.2 PARAMETER SETTINGS.....	9
1.3 CONFIGURATION ITEMS	10
2.0 CLAUSE 1: LOGICAL DATABASE DESIGN	12
2.1 DATABASE DEFINITION STATEMENTS	12
2.2 PHYSICAL ORGANIZATION OF DATABASE	12
2.3 HORIZONTAL PARTITIONING.....	12
2.4 REPLICATION.....	12
3.0 CLAUSE 2: QUERIES AND REFRESH FUNCTIONS RELATED ITEMS	13
3.1 QUERY LANGUAGE	13
3.2 RANDOM NUMBER GENERATION.....	13
3.3 SUBSTITUTION PARAMETERS GENERATION	13
3.4 QUERY TEXT AND OUTPUT DATA FROM DATABASE.....	13
3.5 QUERY SUBSTITUTION PARAMETERS AND SEEDS USED	13
3.6 ISOLATION LEVEL	13
3.7 REFRESH FUNCTIONS	13
4.0 CLAUSE 3: DATABASE SYSTEM PROPERTIES	15
4.1 ATOMICITY REQUIREMENTS	15
4.2 CONSISTENCY REQUIREMENTS	15
4.3 ISOLATION REQUIREMENTS	16
4.4 DURABILITY REQUIREMENTS	17
5.0 CLAUSE 4: SCALING AND DATABASE POPULATION	19
5.1 INITIAL CARDINALITY OF TABLES	19
5.2 DISTRIBUTION OF TABLES AND LOGS ACROSS MEDIA.....	19
5.3 MAPPING OF DATABASE PARTITIONS/REPLICATIONS	20
5.4 IMPLEMENTATION OF RAID	20
5.5 DBGEN MODIFICATIONS	20
5.6 DATABASE LOAD TIME	20
5.7 DATA STORAGE RATIO	20
5.8 DATABASE LOAD MECHANISM DETAILS AND ILLUSTRATION	22
5.9 QUALIFICATION DATABASE CONFIGURATION.....	22
6.0 CLAUSE 5: PERFORMANCE METRICS AND EXECUTION RULES RELATED ITEMS	23
6.1 STEPS IN THE POWER TEST	23
6.2 TIMING INTERVALS FOR EACH QUERY AND REFRESH FUNCTION.....	23
6.3 NUMBER OF STREAMS FOR THE THROUGHPUT TEST	23
6.4 START AND END DATE/TIMES FOR EACH QUERY STREAM	23
6.5 TOTAL ELAPSED TIME FOR THE MEASUREMENT INTERVAL	23
6.6 REFRESH FUNCTION START DATE/TIME AND FINISH DATE/TIME.....	23

6.7	TIMING INTERVALS FOR EACH QUERY AND EACH REFRESH FUNCTION FOR EACH STREAM	23
6.8	PERFORMANCE METRICS	24
6.9	THE PERFORMANCE METRIC AND NUMERICAL QUANTITIES FROM BOTH RUNS	24
6.11	SYSTEM ACTIVITY BETWEEN TESTS	24
7.0	CLAUSE 6: SUT AND DRIVER IMPLEMENTATION RELATED ITEMS	25
7.1	DRIVER	25
7.2	IMPLEMENTATION SPECIFIC LAYER (ISL)	25
7.3	PROFILE-DIRECTED OPTIMIZATION	25
8.0	CLAUSE 7: PRICING RELATED ITEMS	26
8.1	HARDWARE AND SOFTWARE USED	26
8.2	TOTAL 3 YEAR PRICE	26
8.3	AVAILABILITY DATE	26
8.4	COUNTRY-SPECIFIC PRICING	26
9.0	CLAUSE 9: RELATED ITEMS	27
9.1	AUDITORS' REPORT	27
	APPENDIX A: PARAMETER SETTINGS	30
	APPENDIX B: DATABASE BUILD SCRIPTS	34
	APPENDIX C: ACID SCRIPTS	56
	APPENDIX D: QUALIFICATION QUERY TEXT AND OUTPUT	72
	APPENDIX E: SEED AND INPUT PARAMETERS	79
	APPENDIX F: BENCHMARK SCRIPTS	81
	APPENDIX G: PRICE QUOTES	91

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 64P is depicted in Figure 1.1 consists of:

Processors (per blade):	1 x Dual-Core AMD Opteron™ processor Model 285 (2.6GHz/1MB L2 cache per core for 2MB total/1 GHz HyperTransport)
Cores (per blade) :	2
Threads (per blade) :	2
Memory (per blade) :	12 GB
OS Disk Drives (per blade) :	2 x 36GB 15krpm HDD Ultra320
Network (per blade) :	4 x on-board GigE (two of them as cluster interconnect)
Host Bus Adapter (per blade) :	2 x on-board
Storage Area Network :	2 x hp StorageWorks SAN Switch 4/256 129 x hp StorageWorks MSA1000 1806 x 36GB 15krpm HDD Ultra320
Total Storage :	65016GB

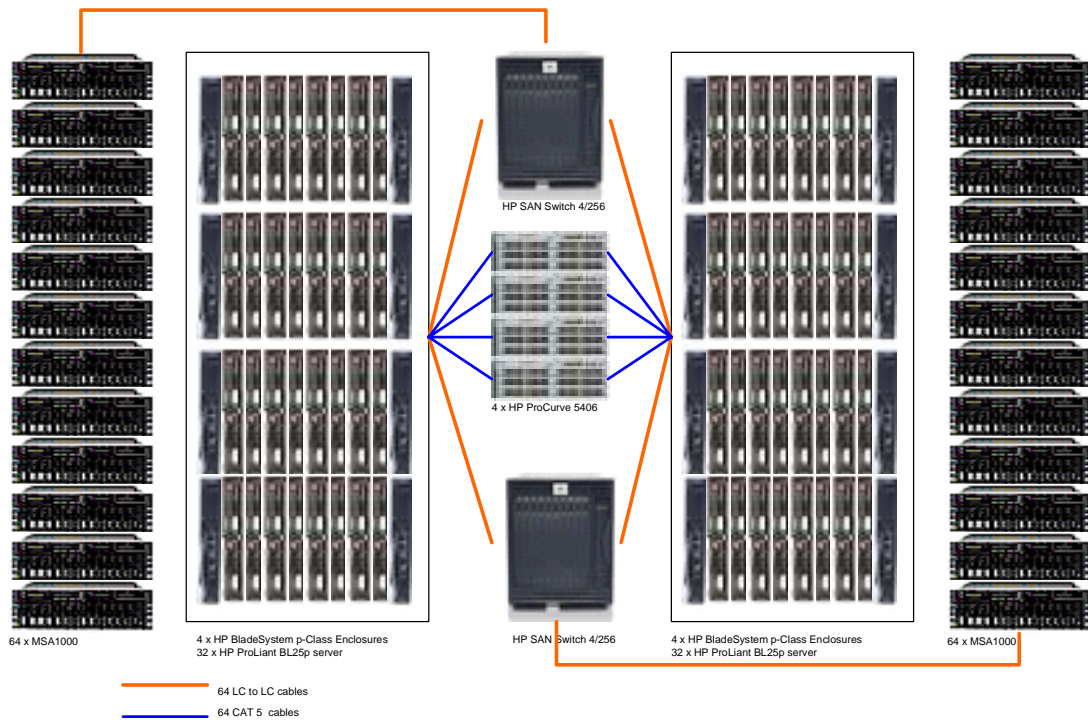
The storage area network (SAN) consists of 2 hp SAN Switch 4/256s and 129 hp StorageWorks MSA1000s. The hp ProLiant BL25p Server Blade has two HBAs, connected to one of two hp SAN Switch 4/256s. One of the hp SAN Switch 4/256s has 64 hp StorageWorks MSA1000s and other one has 65 hp StorageWorks MSA1000s connected to it.

The hp ProLiant BL25p Server Blade has four NICs; one configured for user connectivity, one configured for Oracle Cluster Manager and two as Oracle Cluster Interconnect.

Each hp StorageWorks MSA1000 has one RAID1 volume of 14 x 36GB 15krpm HDD Ultra320s. Each RAID10 volume is partitioned using Linux and hosted database tables, indexes, redo log files. There are 64 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 2.3.0 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 2.3.0 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	450,000,000
Part	600,000,000
Partsupp	2,400,000,000
Orders	4,500,000,000
Lineitem	18,000,048,306

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 64P is depicted in Figure 1.1 consists of:

Processors (per blade):	1 x Dual-Core AMD Opteron™ processor Model 285 (2.6GHz/1MB L2 cache per core for 2MB total/1 GHz HyperTransport)
Cores (per blade) :	2
Threads (per blade) :	2
Memory (per blade) :	12 GB
OS Disk Drives (per blade) :	2 x 36GB 15krpm HDD Ultra320
Network (per blade) :	4 x on-board GigE (two of them as cluster interconnect)
Host Bus Adapter (per blade) :	2 x on-board
Storage Area Network :	2 x hp StorageWorks SAN Switch 4/256 129 x hp StorageWorks MSA1000 1806 x 36GB 15krpm HDD Ultra320
Total Storage :	65016GB

The storage area network (SAN) consists of 2 hp SAN Switch 4/256s and 129 hp StorageWorks MSA1000s. The hp ProLiant BL25p Server Blade has two HBAs, connected to one of two hp SAN Switch 4/256s. One of the hp SAN Switch 4/256s has 64 hp StorageWorks MSA1000s and other one has 65 hp StorageWorks MSA1000s connected to it.

The hp ProLiant BL25p Server Blade has four NICs; one configured for user connectivity, one configured for Oracle Cluster Manager and two as Oracle Cluster Interconnect.

Each hp StorageWorks MSA1000 has one RAID1 volume of 14 x 36GB 15krpm HDD Ultra320s. Each RAID1 volume is partitioned using Linux and hosted database tables, indexes, redo log files. There are 64 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	Disk Drives, Array	Partition	Description	Size
SAN Switch 1 - MSA [1..64] SAN Switch 2 - MSA [65..128]	14 x 36GB, One RAID1 Array of 252GB	1	temp	27GB
		2	tables and indexes	38GB
		3	redo log	1GB
		5	undo	8GB
		6	*misc	512MB
		7	flat files	38GB
		8	redo log	1GB
		SAN Swith 2 - MSA [129]		1,2,3
		*misc -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 1 was used for the entire database and redo log files. Oracle redo file group has two members residing on two separate MSA1000s to guarantee no single point of controller/cache failure.

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 2.3.0 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 5 hours 51 minutes 14 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.

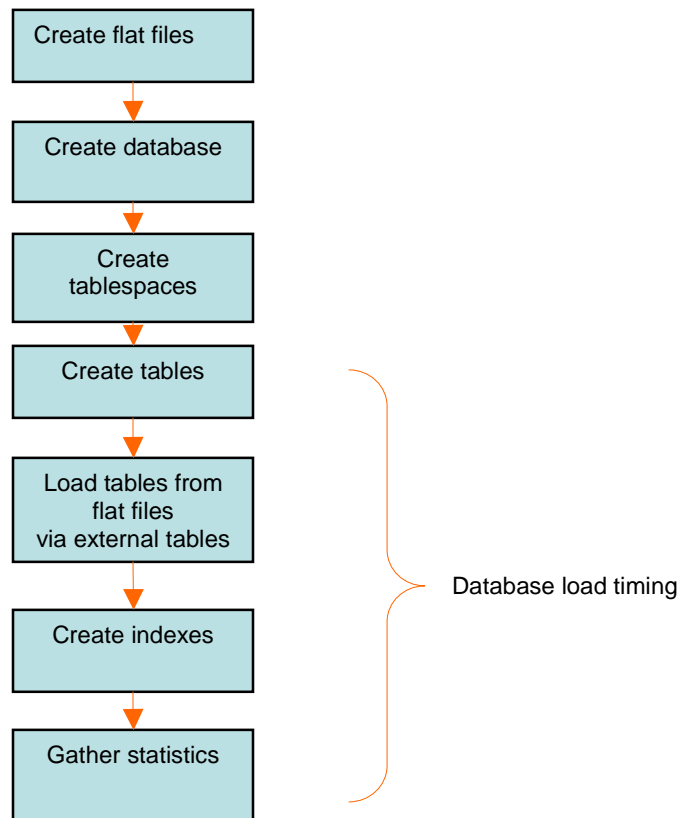
Disk Type	Number of Disks	Total Disk Space	Data Storage Ratio
36GB15krpm HDD Ultra320	1934	69624 GB	23.21

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, and 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.

Eight 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	118,551.0	107,183.9	112,724.3
Run 2	116,379.3	105,063.0	110,576.5

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 activities 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 June 8, 2006.

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.



June 6, 2006

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

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

Platform: ProLiant BL25p 64 node cluster
Database Manager: Oracle Database 10g Enterprise Edition R2
Operating System: Red Hat Enterprise Linux ES

CPU's	Memory	Total Disks	Qpph@ 3000GB	QthH@3000GB	QphH@3000GB
64 AMD Opterons @ 2.6 GHz dual- core	12 GB each node	128 @ 36 GB (OS) 1806 @ 36GB	116,379.3	105,063.0	110,576.5

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

- The database tables were defined with the proper columns, layout and sizes.
- The tested database was correctly scaled and populated for 3000GB using DBGEN. The version of DBGEN was 2.3.0.
- 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 2.3.0.
- 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,

A handwritten signature in cursive script that reads "Lorna Livingtree".

Lorna Livingtree
Auditor

Appendix A: Parameter Settings

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

. $FRAME_PATH/env
echo $START_NODES
for i in $START_NODES
do
echo starting instance on node $i
ssh $i -n /home/oracle/frame/bin/start &
done

----- init_bigh101.ora -----
instance_number = 1
thread = 1
undo_management = auto
undo_tablespace = ts_undo2
cluster_database = true
cluster_interconnects = 2.2.2.101
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh102.ora -----
instance_number = 2
thread = 2
undo_management = auto
undo_tablespace = ts_undo2
cluster_database = true
cluster_interconnects = 3.3.3.102
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh103.ora -----
instance_number = 3
thread = 3
undo_management = auto
undo_tablespace = ts_undo3
cluster_database = true
cluster_interconnects = 2.2.2.103
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh104.ora -----
instance_number = 4
thread = 4
undo_management = auto
undo_tablespace = ts_undo4
cluster_database = true
cluster_interconnects = 3.3.3.104
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh105.ora -----
instance_number = 5
thread = 5
undo_management = auto
undo_tablespace = ts_undo5
cluster_database = true
cluster_interconnects = 2.2.2.105
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh106.ora -----
instance_number = 6
thread = 6
undo_management = auto
undo_tablespace = ts_undo6
cluster_database = true
cluster_interconnects = 3.3.3.106
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh107.ora -----
instance_number = 7
thread = 7
undo_management = auto
undo_tablespace = ts_undo7
cluster_database = true
cluster_interconnects = 2.2.2.107
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh108.ora -----
instance_number = 8
thread = 8
undo_management = auto
undo_tablespace = ts_undo8
cluster_database = true
cluster_interconnects = 3.3.3.108
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh109.ora -----
instance_number = 9
thread = 9
undo_management = auto
undo_tablespace = ts_undo9
cluster_database = true
cluster_interconnects = 2.2.2.109
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh110.ora -----
```

```
instance_number = 10
thread = 10
undo_management = auto
undo_tablespace = ts_undo10
cluster_database = true
cluster_interconnects = 3.3.3.110
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh111.ora -----
instance_number = 11
thread = 11
undo_management = auto
undo_tablespace = ts_undo11
cluster_database = true
cluster_interconnects = 2.2.2.111
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh112.ora -----
instance_number = 12
thread = 12
undo_management = auto
undo_tablespace = ts_undo12
cluster_database = true
cluster_interconnects = 3.3.3.112
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh113.ora -----
instance_number = 13
thread = 13
undo_management = auto
undo_tablespace = ts_undo13
cluster_database = true
cluster_interconnects = 2.2.2.113
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh114.ora -----
instance_number = 14
thread = 14
undo_management = auto
undo_tablespace = ts_undo14
cluster_database = true
cluster_interconnects = 3.3.3.114
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh115.ora -----
instance_number = 15
thread = 15
undo_management = auto
undo_tablespace = ts_undo15
cluster_database = true
cluster_interconnects = 2.2.2.115
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh116.ora -----
instance_number = 16
thread = 16
undo_management = auto
undo_tablespace = ts_undo16
cluster_database = true
cluster_interconnects = 3.3.3.116
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh117.ora -----
instance_number = 17
thread = 17
undo_management = auto
undo_tablespace = ts_undo17
cluster_database = true
cluster_interconnects = 2.2.2.117
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh118.ora -----
instance_number = 18
thread = 18
undo_management = auto
undo_tablespace = ts_undo18
cluster_database = true
cluster_interconnects = 3.3.3.118
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh119.ora -----
instance_number = 19
thread = 19
undo_management = auto
undo_tablespace = ts_undo19
cluster_database = true
cluster_interconnects = 2.2.2.119
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh120.ora -----
instance_number = 20
thread = 20
undo_management = auto
undo_tablespace = ts_undo20
```



```

ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh144.ora -----
instance_number = 44
thread = 44
undo_management = auto
undo_tablespace = ts_undo44
cluster_database = true
cluster_interconnects = 3.3.3.144
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh145.ora -----
instance_number = 45
thread = 45
undo_management = auto
undo_tablespace = ts_undo45
cluster_database = true
cluster_interconnects = 2.2.2.145
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh146.ora -----
instance_number = 46
thread = 46
undo_management = auto
undo_tablespace = ts_undo46
cluster_database = true
cluster_interconnects = 3.3.3.146
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh147.ora -----
instance_number = 47
thread = 47
undo_management = auto
undo_tablespace = ts_undo47
cluster_database = true
cluster_interconnects = 2.2.2.147
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh148.ora -----
instance_number = 48
thread = 48
undo_management = auto
undo_tablespace = ts_undo48
cluster_database = true
cluster_interconnects = 3.3.3.148
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh149.ora -----
instance_number = 49
thread = 49
undo_management = auto
undo_tablespace = ts_undo49
cluster_database = true
cluster_interconnects = 2.2.2.149
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh150.ora -----
instance_number = 50
thread = 50
undo_management = auto
undo_tablespace = ts_undo50
cluster_database = true
cluster_interconnects = 3.3.3.150
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh151.ora -----
instance_number = 51
thread = 51
undo_management = auto
undo_tablespace = ts_undo51
cluster_database = true
cluster_interconnects = 2.2.2.151
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh152.ora -----
instance_number = 52
thread = 52
undo_management = auto
undo_tablespace = ts_undo52
cluster_database = true
cluster_interconnects = 3.3.3.152
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh153.ora -----
instance_number = 53
thread = 53
undo_management = auto
undo_tablespace = ts_undo53
cluster_database = true
cluster_interconnects = 2.2.2.153
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh154.ora -----
instance_number = 54
thread = 54
undo_management = auto
undo_tablespace = ts_undo54
cluster_database = true
cluster_interconnects = 3.3.3.154
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh155.ora -----
instance_number = 55
thread = 55
undo_management = auto

```

```

undo_tablespace = ts_undo55
cluster_database = true
cluster_interconnects = 2.2.2.155
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh156.ora -----
instance_number = 56
thread = 56
undo_management = auto
undo_tablespace = ts_undo56
cluster_database = true
cluster_interconnects = 3.3.3.156
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh157.ora -----
instance_number = 57
thread = 57
undo_management = auto
undo_tablespace = ts_undo57
cluster_database = true
cluster_interconnects = 2.2.2.157
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh158.ora -----
instance_number = 58
thread = 58
undo_management = auto
undo_tablespace = ts_undo58
cluster_database = true
cluster_interconnects = 3.3.3.158
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh159.ora -----
instance_number = 59
thread = 59
undo_management = auto
undo_tablespace = ts_undo59
cluster_database = true
cluster_interconnects = 2.2.2.159
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh160.ora -----
instance_number = 60
thread = 60
undo_management = auto
undo_tablespace = ts_undo60
cluster_database = true
cluster_interconnects = 3.3.3.160
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh161.ora -----
instance_number = 61
thread = 61
undo_management = auto
undo_tablespace = ts_undo61
cluster_database = true
cluster_interconnects = 2.2.2.161
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh162.ora -----
instance_number = 62
thread = 62
undo_management = auto
undo_tablespace = ts_undo62
cluster_database = true
cluster_interconnects = 3.3.3.162
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh163.ora -----
instance_number = 63
thread = 63
undo_management = auto
undo_tablespace = ts_undo63
cluster_database = true
cluster_interconnects = 2.2.2.163
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_bigh164.ora -----
instance_number = 64
thread = 64
undo_management = auto
undo_tablespace = ts_undo64
cluster_database = true
cluster_interconnects = 3.3.3.164
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
----- init_build.ora -----
aq_tm_processes = 0
audit_trail = false
compatible = 10.2.0.1
control_files = (/home/oracle/dev/block/control1,/home/oracle/dev/block/control2)
db_block_checksum = false
db_block_size = 16384
db_cache_size = 2000m
db_file_multiblock_read_count = 48
db_files = 500
db_name = 10i
db_writer_processes = 4
dml_locks = 5000
global_names = false
instance_name = raca
log_buffer = 4194304
log_checkpoint_timeout = 1200

```

```

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 = 16384
parallel_max_servers    = 100
parallel_min_servers    = 48
pga_aggregate_target   = 5500m
processes               = 1000
recovery_parallelism    = 8
replication_dependency_tracking = false
statistics_level        = basic
undo_retention          = 400000
optimizer_index_cost_adj = 1400
cpu_count               = 2
shared_pool_size        = 2000m
----- rawdevices -----
/home/oracle/dev/raw/ocr1 /dev/sddy1
/home/oracle/dev/raw/ocr2 /dev/sddy2
/home/oracle/dev/raw/quorum1 /dev/sddy3
#/home/oracle/dev/raw/ocr1 /dev/sda8
#/home/oracle/dev/raw/ocr2 /dev/sdb8
#/home/oracle/dev/raw/quorum1 /dev/sdc8
#/home/oracle/dev/raw/quorum2 /dev/sdd8
#/home/oracle/dev/raw/quorum3 /dev/sde8
----- 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
#insmod /lib/modules/2.6.9-32.ELsmp/kernel/drivers/addon/oracleasm/oracleasm.ko
#sh /etc/oracle-asm-start.sh
#chown -R oracle:oracle /home/oracle/dev/*
#rmmod bnx2
chown -R oracle:oracle /home/oracle/dev/*

#changes from Raghu Saturday April 29th
#ifconfig eth0 txqueuelen 2000
ifconfig eth1 txqueuelen 2000
ifconfig eth2 txqueuelen 2000

sh /etc/mount-stats-dir.sh
----- rc.local.node2-64 -----
#!/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
#insmod /lib/modules/2.6.9-32.ELsmp/kernel/drivers/addon/oracleasm/oracleasm.ko
#sh /etc/oracle-asm-start.sh
#chown -R oracle:oracle /home/oracle/dev/*
#rmmod bnx2
chown -R oracle:oracle /home/oracle/dev/*

#changes from Raghu Saturday April 29th
#ifconfig eth0 txqueuelen 2000
ifconfig eth1 txqueuelen 2000
ifconfig eth2 txqueuelen 2000

#sh /etc/mount-stats-dir.sh
----- 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

fs.file-max = 4194304

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
#net.ipv4.ip_local_port_range = 32768 65000

#net.core.rmem_max = 2097152
#net.core.rmem_default = 2097152
#net.core.wmem_max = 2097152
#net.core.wmem_default = 2097152
#net.ipv4.ip_local_port_range = 32768 65000

#fs.aio-max-nr = 2097152
fs.aio-max-nr = 4194304
#kernel.shmmax = 4000000000
kernel.shmmax = 4563402752

#2.6 RKO
net.ipv4.ip_local_port_range = 32768 65000
net.core.rmem_default = 2097152
net.core.wmem_default = 2097152
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
net.ipv4.tcp_no_metrics_save = 1
net.ipv4.tcp_max_syn_backlog = 1536
net.core.netdev_max_backlog = 3000
net.ipv4.tcp_timestamps = 0
net.ipv4.tcp_sack = 1
net.ipv4.tcp_window_scaling = 1
----- tstart -----
#!/bin/ksh
#
# SHeader: 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
startup $PFILE

exit
!

cd $DIR

```

Appendix B: Database Build Scripts

```

----- addts.sh -----
sqlplus /NOLOG<<!
connect / as sysdba;
alter tablespace $1 add datafile '$2' size $3 reuse;
!

----- addtts.sh -----
sqlplus /NOLOG<<!
connect / as sysdba;
alter tablespace $1 add tempfile '$2' size $3 reuse;
!

----- addundo.log.sh -----
one=$1
((one=one*2-1))
((two=one+1))
echo start creating undo and log for node $1 `date`
sqlplus /NOLOG <<!
connect / as sysdba;
create undo tablespace ts_undo${1} datafile '/home/oracle/dev/block/undo_${1}' size 1536m
reuse;
alter database add logfile thread ${1} '/home/oracle/dev/block/log_${one}' size 1024m reuse,
'/home/oracle/dev/block/log_${two}' size 1024m reuse,
'/home/oracle/dev/block/log_${one}_a' size 1024m reuse,
'/home/oracle/dev/block/log_${two}_a' size 1024m reuse;
alter database enable public thread ${1};
!
echo end creating undo and log for node $1 `date`
----- createdire.sh -----
#!/bin/bash
(( dir=0 ))
while (( dir<128 ));do
  (( dir=dir+1 ))
  sqlplus /NOLOG <<EOF
  connect tpch/tpch;
  drop directory ff${dir};
  create directory ff${dir} as '/home/oracle/dev/ff_${dir}';
  drop directory ffu${dir};
  create directory ffu${dir} as '/home/oracle/dev/ff_${dir}/update_sets';
  EOF
done
----- create_et4rf.sh -----
#!/bin/ksh
((SETNUM=0))
while ((SETNUM<20));do
  ((SETNUM=SETNUM+1))
  sqlplus /NOLOG << !
  connect tpch/tpch;
  set timing on
  set serveroutput on
  set echo on

  drop table temp_l_et${SETNUM};
  create table temp_l_et${SETNUM}(
    l_orderkey      number ,
    l_partkey       number ,
    l_suppkey        number ,
    l_linenum       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 ffu1
    access parameters
    (
      records delimited by newline
      nobadfile
      nologfile
      fields terminated by "|"
      missing field values are null
    )
    location (
    ffu1:'lineitem.tbl.u${SETNUM}.1',
    ffu2:'lineitem.tbl.u${SETNUM}.2',
    ffu3:'lineitem.tbl.u${SETNUM}.3',
    ffu4:'lineitem.tbl.u${SETNUM}.4',
    ffu5:'lineitem.tbl.u${SETNUM}.5',
    ffu6:'lineitem.tbl.u${SETNUM}.6',
    ffu7:'lineitem.tbl.u${SETNUM}.7',
    ffu8:'lineitem.tbl.u${SETNUM}.8',
    ffu9:'lineitem.tbl.u${SETNUM}.9',
    ffu10:'lineitem.tbl.u${SETNUM}.10',
    ffu11:'lineitem.tbl.u${SETNUM}.11',
    ffu12:'lineitem.tbl.u${SETNUM}.12',
    ffu13:'lineitem.tbl.u${SETNUM}.13',
    ffu14:'lineitem.tbl.u${SETNUM}.14',
    ffu15:'lineitem.tbl.u${SETNUM}.15',
    ffu16:'lineitem.tbl.u${SETNUM}.16',
    ffu17:'lineitem.tbl.u${SETNUM}.17',
    ffu18:'lineitem.tbl.u${SETNUM}.18',
    ffu19:'lineitem.tbl.u${SETNUM}.19',
    ffu20:'lineitem.tbl.u${SETNUM}.20',
    ffu21:'lineitem.tbl.u${SETNUM}.21',
    ffu22:'lineitem.tbl.u${SETNUM}.22',
    ffu23:'lineitem.tbl.u${SETNUM}.23',
    ffu24:'lineitem.tbl.u${SETNUM}.24',
    ffu25:'lineitem.tbl.u${SETNUM}.25',
    ffu26:'lineitem.tbl.u${SETNUM}.26',
    ffu27:'lineitem.tbl.u${SETNUM}.27',
    ffu28:'lineitem.tbl.u${SETNUM}.28',
    ffu29:'lineitem.tbl.u${SETNUM}.29',
    ffu30:'lineitem.tbl.u${SETNUM}.30',
    ffu31:'lineitem.tbl.u${SETNUM}.31',
    ffu32:'lineitem.tbl.u${SETNUM}.32',
    ffu33:'lineitem.tbl.u${SETNUM}.33',
    ffu34:'lineitem.tbl.u${SETNUM}.34',
    ffu35:'lineitem.tbl.u${SETNUM}.35',
    ffu36:'lineitem.tbl.u${SETNUM}.36',
    ffu37:'lineitem.tbl.u${SETNUM}.37',
    ffu38:'lineitem.tbl.u${SETNUM}.38',
    ffu39:'lineitem.tbl.u${SETNUM}.39',
    ffu40:'lineitem.tbl.u${SETNUM}.40',
    ffu41:'lineitem.tbl.u${SETNUM}.41',
    ffu42:'lineitem.tbl.u${SETNUM}.42',
    ffu43:'lineitem.tbl.u${SETNUM}.43',
    ffu44:'lineitem.tbl.u${SETNUM}.44',
    ffu45:'lineitem.tbl.u${SETNUM}.45',
    ffu46:'lineitem.tbl.u${SETNUM}.46',
    ffu47:'lineitem.tbl.u${SETNUM}.47',
    ffu48:'lineitem.tbl.u${SETNUM}.48',
    ffu49:'lineitem.tbl.u${SETNUM}.49',
    ffu50:'lineitem.tbl.u${SETNUM}.50',
    ffu51:'lineitem.tbl.u${SETNUM}.51',
    ffu52:'lineitem.tbl.u${SETNUM}.52',
    ffu53:'lineitem.tbl.u${SETNUM}.53',
    ffu54:'lineitem.tbl.u${SETNUM}.54',
    ffu55:'lineitem.tbl.u${SETNUM}.55',
    ffu56:'lineitem.tbl.u${SETNUM}.56',
    ffu57:'lineitem.tbl.u${SETNUM}.57',
    ffu58:'lineitem.tbl.u${SETNUM}.58',
    ffu59:'lineitem.tbl.u${SETNUM}.59',
    ffu60:'lineitem.tbl.u${SETNUM}.60',
    ffu61:'lineitem.tbl.u${SETNUM}.61',
    ffu62:'lineitem.tbl.u${SETNUM}.62',
    ffu63:'lineitem.tbl.u${SETNUM}.63',
    ffu64:'lineitem.tbl.u${SETNUM}.64'
  )
  reject limit unlimited parallel 64;

  drop table temp_o_et${SETNUM};
  create table temp_o_et${SETNUM}(
    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 (
ffu1:'orders.tbl.us${SETNUM}.1',
ffu2:'orders.tbl.us${SETNUM}.2',
ffu3:'orders.tbl.us${SETNUM}.3',
ffu4:'orders.tbl.us${SETNUM}.4',
ffu5:'orders.tbl.us${SETNUM}.5',
ffu6:'orders.tbl.us${SETNUM}.6',
ffu7:'orders.tbl.us${SETNUM}.7',
ffu8:'orders.tbl.us${SETNUM}.8',
ffu9:'orders.tbl.us${SETNUM}.9',
ffu10:'orders.tbl.us${SETNUM}.10',
ffu11:'orders.tbl.us${SETNUM}.11',
ffu12:'orders.tbl.us${SETNUM}.12',
ffu13:'orders.tbl.us${SETNUM}.13',
ffu14:'orders.tbl.us${SETNUM}.14',
ffu15:'orders.tbl.us${SETNUM}.15',
ffu16:'orders.tbl.us${SETNUM}.16',
ffu17:'orders.tbl.us${SETNUM}.17',
ffu18:'orders.tbl.us${SETNUM}.18',
ffu19:'orders.tbl.us${SETNUM}.19',
ffu20:'orders.tbl.us${SETNUM}.20',
ffu21:'orders.tbl.us${SETNUM}.21',
ffu22:'orders.tbl.us${SETNUM}.22',
ffu23:'orders.tbl.us${SETNUM}.23',
ffu24:'orders.tbl.us${SETNUM}.24',
ffu25:'orders.tbl.us${SETNUM}.25',
ffu26:'orders.tbl.us${SETNUM}.26',
ffu27:'orders.tbl.us${SETNUM}.27',
ffu28:'orders.tbl.us${SETNUM}.28',
ffu29:'orders.tbl.us${SETNUM}.29',
ffu30:'orders.tbl.us${SETNUM}.30',
ffu31:'orders.tbl.us${SETNUM}.31',
ffu32:'orders.tbl.us${SETNUM}.32',
ffu33:'orders.tbl.us${SETNUM}.33',
ffu34:'orders.tbl.us${SETNUM}.34',
ffu35:'orders.tbl.us${SETNUM}.35',
ffu36:'orders.tbl.us${SETNUM}.36',
ffu37:'orders.tbl.us${SETNUM}.37',
ffu38:'orders.tbl.us${SETNUM}.38',
ffu39:'orders.tbl.us${SETNUM}.39',
ffu40:'orders.tbl.us${SETNUM}.40',
ffu41:'orders.tbl.us${SETNUM}.41',
ffu42:'orders.tbl.us${SETNUM}.42',
ffu43:'orders.tbl.us${SETNUM}.43',
ffu44:'orders.tbl.us${SETNUM}.44',
ffu45:'orders.tbl.us${SETNUM}.45',
ffu46:'orders.tbl.us${SETNUM}.46',
ffu47:'orders.tbl.us${SETNUM}.47',
ffu48:'orders.tbl.us${SETNUM}.48',
ffu49:'orders.tbl.us${SETNUM}.49',
ffu50:'orders.tbl.us${SETNUM}.50',
ffu51:'orders.tbl.us${SETNUM}.51',
ffu52:'orders.tbl.us${SETNUM}.52',
ffu53:'orders.tbl.us${SETNUM}.53',
ffu54:'orders.tbl.us${SETNUM}.54',
ffu55:'orders.tbl.us${SETNUM}.55',
ffu56:'orders.tbl.us${SETNUM}.56',
ffu57:'orders.tbl.us${SETNUM}.57',
ffu58:'orders.tbl.us${SETNUM}.58',
ffu59:'orders.tbl.us${SETNUM}.59',
ffu60:'orders.tbl.us${SETNUM}.60',
ffu61:'orders.tbl.us${SETNUM}.61',
ffu62:'orders.tbl.us${SETNUM}.62',
ffu63:'orders.tbl.us${SETNUM}.63',
ffu64:'orders.tbl.us${SETNUM}.64'
))
reject limit unlimited parallel 64;

drop table temp_okey_et${SETNUM};
create table temp_okey_et${SETNUM}(
  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 (
ffu1:'delete.us${SETNUM}.1',
ffu2:'delete.us${SETNUM}.2',
ffu3:'delete.us${SETNUM}.3',
ffu4:'delete.us${SETNUM}.4',
ffu5:'delete.us${SETNUM}.5',
ffu6:'delete.us${SETNUM}.6',

```

```

ffu7:'delete.us${SETNUM}.7',
ffu8:'delete.us${SETNUM}.8',
ffu9:'delete.us${SETNUM}.9',
ffu10:'delete.us${SETNUM}.10',
ffu11:'delete.us${SETNUM}.11',
ffu12:'delete.us${SETNUM}.12',
ffu13:'delete.us${SETNUM}.13',
ffu14:'delete.us${SETNUM}.14',
ffu15:'delete.us${SETNUM}.15',
ffu16:'delete.us${SETNUM}.16',
ffu17:'delete.us${SETNUM}.17',
ffu18:'delete.us${SETNUM}.18',
ffu19:'delete.us${SETNUM}.19',
ffu20:'delete.us${SETNUM}.20',
ffu21:'delete.us${SETNUM}.21',
ffu22:'delete.us${SETNUM}.22',
ffu23:'delete.us${SETNUM}.23',
ffu24:'delete.us${SETNUM}.24',
ffu25:'delete.us${SETNUM}.25',
ffu26:'delete.us${SETNUM}.26',
ffu27:'delete.us${SETNUM}.27',
ffu28:'delete.us${SETNUM}.28',
ffu29:'delete.us${SETNUM}.29',
ffu30:'delete.us${SETNUM}.30',
ffu31:'delete.us${SETNUM}.31',
ffu32:'delete.us${SETNUM}.32',
ffu33:'delete.us${SETNUM}.33',
ffu34:'delete.us${SETNUM}.34',
ffu35:'delete.us${SETNUM}.35',
ffu36:'delete.us${SETNUM}.36',
ffu37:'delete.us${SETNUM}.37',
ffu38:'delete.us${SETNUM}.38',
ffu39:'delete.us${SETNUM}.39',
ffu40:'delete.us${SETNUM}.40',
ffu41:'delete.us${SETNUM}.41',
ffu42:'delete.us${SETNUM}.42',
ffu43:'delete.us${SETNUM}.43',
ffu44:'delete.us${SETNUM}.44',
ffu45:'delete.us${SETNUM}.45',
ffu46:'delete.us${SETNUM}.46',
ffu47:'delete.us${SETNUM}.47',
ffu48:'delete.us${SETNUM}.48',
ffu49:'delete.us${SETNUM}.49',
ffu50:'delete.us${SETNUM}.50',
ffu51:'delete.us${SETNUM}.51',
ffu52:'delete.us${SETNUM}.52',
ffu53:'delete.us${SETNUM}.53',
ffu54:'delete.us${SETNUM}.54',
ffu55:'delete.us${SETNUM}.55',
ffu56:'delete.us${SETNUM}.56',
ffu57:'delete.us${SETNUM}.57',
ffu58:'delete.us${SETNUM}.58',
ffu59:'delete.us${SETNUM}.59',
ffu60:'delete.us${SETNUM}.60',
ffu61:'delete.us${SETNUM}.61',
ffu62:'delete.us${SETNUM}.62',
ffu63:'delete.us${SETNUM}.63',
ffu64:'delete.us${SETNUM}.64'
))
reject limit unlimited parallel 4;
!

done
----- crts.sh -----
i=$1;
sqlplus /NOLOG<<!
connect / as sysdba;
drop tablespace tsd${i} including contents;
create tablespace tsd${i} nologging
datafile '/home/oracle/dev/block/lo_${i}' size ${2} reuse extent management dictionary default
storage (initial 25m next 10m maxextents unlimited pctincrease 0);
!

----- dapop_AUDIT.sh -----
#!/bin/ksh
. $KIT_DIR/env

echo Start Load at `date`

sqlplus /NOLOG <<EOF
connect / as sysdba
drop user tpch cascade;
grant DBA
to tpch identified by tpch;
EOF

/home/oracle/kit/audit/createdire.sh
/home/oracle/kit/audit/create_et4rf.sh

((i=0))
while ((i<64)) ; do
  ((i=i+1))
  sleep 10

```

```

/home/oracle/kit/audit/dapopo_AUDIT.sh $i &
done
wait

((i=64))
while ((i<84)) ; do
  ((i=i+1))
  sleep 10
  /home/oracle/kit/audit/dapopo_AUDIT.sh $i &
done

((i=0))
while ((i<44)) ; do
  ((i=i+1))
  sleep 10
  /home/oracle/kit/audit/dapopl_AUDIT.sh $i &
done
wait
wait

((i=44))
while ((i<84)) ; do
  ((i=i+1))
  sleep 10
  /home/oracle/kit/audit/dapopl_AUDIT.sh $i &
done

wait
wait

/home/oracle/kit/audit/ex_linparts.sh
/home/oracle/kit/audit/ex_ordparts.sh
/home/oracle/kit/audit/dapopre_AUDIT.sh
wait
sqlplus /NOLOG << EOF
connect tpch/tpch
insert into lineitem (select * from lineitem_item84);
drop table lineitem_item84;
set timing on
alter table lineitem parallel;
alter table orders parallel;
alter table part parallel;
alter table supplier parallel;
alter table partsupp parallel;
alter table customer parallel;
alter index index_i_l_orderkey parallel;
alter index i_o_orderkey parallel;
alter index i_c_custkey parallel;
alter index i_ps_partkey_supkey parallel;
connect / as sysdba
execute dbms_stats.gather_schema_stats('TPCH', estimate_percent => 1, degree => 256,
granularity => 'GLOBAL');
execute dbms_stats.gather_system_stats;
execute dbms_scheduler.disable('GATHER_STATS_JOB');
execute dbms_scheduler.disable('AUTO_SPACE_ADVISOR_JOB');
execute dbms_scheduler.disable('AUTO_TASKS_JOB_CLASS');
alter system switch logfile;
!date
EOF
echo End Load at `date`
----- dapopre_AUDIT.sh -----
#!/bin/ksh
i=$1
sqlplus /NOLOG <<!! &

connect tpch/tpch
drop table l_et_$(i);
create table l_et_$(i)(
  l_orderkey      number ,
  l_partkey       number ,
  l_supkey        number ,
  l_linenum       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 ffl
access parameters
(
  records delimited by newline
  nobadfile
  nologle

```

```

fields terminated by '|'
missing field values are null
)
location (
ff1:'lineitem.tbl_$(i)', ff2:'lineitem.tbl_$(i)', ff3:'lineitem.tbl_$(i)', ff4:'lineitem.tbl_$(i)',
ff5:'lineitem.tbl_$(i)', ff6:'lineitem.tbl_$(i)', ff7:'lineitem.tbl_$(i)', ff8:'lineitem.tbl_$(i)',
ff9:'lineitem.tbl_$(i)', ff10:'lineitem.tbl_$(i)', ff11:'lineitem.tbl_$(i)', ff12:'lineitem.tbl_$(i)',
ff13:'lineitem.tbl_$(i)', ff14:'lineitem.tbl_$(i)', ff15:'lineitem.tbl_$(i)', ff16:'lineitem.tbl_$(i)',
ff17:'lineitem.tbl_$(i)', ff18:'lineitem.tbl_$(i)', ff19:'lineitem.tbl_$(i)', ff20:'lineitem.tbl_$(i)',
ff21:'lineitem.tbl_$(i)', ff22:'lineitem.tbl_$(i)', ff23:'lineitem.tbl_$(i)', ff24:'lineitem.tbl_$(i)',
ff25:'lineitem.tbl_$(i)', ff26:'lineitem.tbl_$(i)', ff27:'lineitem.tbl_$(i)', ff28:'lineitem.tbl_$(i)',
ff29:'lineitem.tbl_$(i)', ff30:'lineitem.tbl_$(i)', ff31:'lineitem.tbl_$(i)', ff32:'lineitem.tbl_$(i)',
ff33:'lineitem.tbl_$(i)', ff34:'lineitem.tbl_$(i)', ff35:'lineitem.tbl_$(i)', ff36:'lineitem.tbl_$(i)',
ff37:'lineitem.tbl_$(i)', ff38:'lineitem.tbl_$(i)', ff39:'lineitem.tbl_$(i)', ff40:'lineitem.tbl_$(i)',
ff41:'lineitem.tbl_$(i)', ff42:'lineitem.tbl_$(i)', ff43:'lineitem.tbl_$(i)', ff44:'lineitem.tbl_$(i)',
ff45:'lineitem.tbl_$(i)', ff46:'lineitem.tbl_$(i)', ff47:'lineitem.tbl_$(i)', ff48:'lineitem.tbl_$(i)',
ff49:'lineitem.tbl_$(i)', ff50:'lineitem.tbl_$(i)', ff51:'lineitem.tbl_$(i)', ff52:'lineitem.tbl_$(i)',
ff53:'lineitem.tbl_$(i)', ff54:'lineitem.tbl_$(i)', ff55:'lineitem.tbl_$(i)', ff56:'lineitem.tbl_$(i)',
ff57:'lineitem.tbl_$(i)', ff58:'lineitem.tbl_$(i)', ff59:'lineitem.tbl_$(i)', ff60:'lineitem.tbl_$(i)',
ff61:'lineitem.tbl_$(i)', ff62:'lineitem.tbl_$(i)', ff63:'lineitem.tbl_$(i)', ff64:'lineitem.tbl_$(i)',
ff65:'lineitem.tbl_$(i)', ff66:'lineitem.tbl_$(i)', ff67:'lineitem.tbl_$(i)', ff68:'lineitem.tbl_$(i)',
ff69:'lineitem.tbl_$(i)', ff70:'lineitem.tbl_$(i)', ff71:'lineitem.tbl_$(i)', ff72:'lineitem.tbl_$(i)',
ff73:'lineitem.tbl_$(i)', ff74:'lineitem.tbl_$(i)', ff75:'lineitem.tbl_$(i)', ff76:'lineitem.tbl_$(i)',
ff77:'lineitem.tbl_$(i)', ff78:'lineitem.tbl_$(i)', ff79:'lineitem.tbl_$(i)', ff80:'lineitem.tbl_$(i)',
ff81:'lineitem.tbl_$(i)', ff82:'lineitem.tbl_$(i)', ff83:'lineitem.tbl_$(i)', ff84:'lineitem.tbl_$(i)',
ff85:'lineitem.tbl_$(i)', ff86:'lineitem.tbl_$(i)', ff87:'lineitem.tbl_$(i)', ff88:'lineitem.tbl_$(i)',
ff89:'lineitem.tbl_$(i)', ff90:'lineitem.tbl_$(i)', ff91:'lineitem.tbl_$(i)', ff92:'lineitem.tbl_$(i)',
ff93:'lineitem.tbl_$(i)', ff94:'lineitem.tbl_$(i)', ff95:'lineitem.tbl_$(i)', ff96:'lineitem.tbl_$(i)',
ff97:'lineitem.tbl_$(i)', ff98:'lineitem.tbl_$(i)', ff99:'lineitem.tbl_$(i)', ff100:'lineitem.tbl_$(i)',
ff101:'lineitem.tbl_$(i)', ff102:'lineitem.tbl_$(i)', ff103:'lineitem.tbl_$(i)',
ff104:'lineitem.tbl_$(i)',
ff105:'lineitem.tbl_$(i)', ff106:'lineitem.tbl_$(i)', ff107:'lineitem.tbl_$(i)',
ff108:'lineitem.tbl_$(i)',
ff109:'lineitem.tbl_$(i)', ff110:'lineitem.tbl_$(i)', ff111:'lineitem.tbl_$(i)',
ff112:'lineitem.tbl_$(i)',
ff113:'lineitem.tbl_$(i)', ff114:'lineitem.tbl_$(i)', ff115:'lineitem.tbl_$(i)',
ff116:'lineitem.tbl_$(i)',
ff117:'lineitem.tbl_$(i)', ff118:'lineitem.tbl_$(i)', ff119:'lineitem.tbl_$(i)',
ff120:'lineitem.tbl_$(i)',
ff121:'lineitem.tbl_$(i)', ff122:'lineitem.tbl_$(i)', ff123:'lineitem.tbl_$(i)',
ff124:'lineitem.tbl_$(i)',
ff125:'lineitem.tbl_$(i)', ff126:'lineitem.tbl_$(i)', ff127:'lineitem.tbl_$(i)',
ff128:'lineitem.tbl_$(i)'
))reject limit unlimited;

alter table l_et_$(i) parallel 4;

alter session force parallel dml parallel (degree 4);
set timing on
set echo on

!echo start loading table lineitem_item$(i)
!date

drop table lineitem_item$(i);
create table lineitem_item$(i)(
  l_shipdate      ,
  l_orderkey      NOT NULL,
  l_discount      NOT NULL,
  l_extendedprice NOT NULL,
  l_supkey        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_linenum       NOT NULL,
  l_shipinstruct  ,
  l_comment       ,
)
pctfree 1
pctused 99
intrans 10
storage (initial 1m next 16m pctincrease 0 freelist groups 8 freelists 99)
compress
parallel 4
nologging
partition by hash(l_partkey)
partitions 256
store in ( tsd1.tsd2.tsd3.tsd4.tsd5.tsd6.tsd7.tsd8.tsd9.tsd10.tsd11,
tsd12.tsd13.tsd14.tsd15.tsd16.tsd17.tsd18.tsd19.tsd20.tsd21.tsd22,
tsd23.tsd24.tsd25.tsd26.tsd27.tsd28.tsd29.tsd30.tsd31.tsd32.tsd33,
tsd34.tsd35.tsd36.tsd37.tsd38.tsd39.tsd40.tsd41.tsd42.tsd43.tsd44,
tsd45.tsd46.tsd47.tsd48.tsd49.tsd50.tsd51.tsd52.tsd53.tsd54.tsd55,
tsd56.tsd57.tsd58.tsd59.tsd60.tsd61.tsd62.tsd63.tsd64.tsd65.tsd66,
tsd67.tsd68.tsd69.tsd70.tsd71.tsd72.tsd73.tsd74.tsd75.tsd76.tsd77,
tsd78.tsd79.tsd80.tsd81.tsd82.tsd83.tsd84.tsd85.tsd86.tsd87.tsd88,
tsd89.tsd90.tsd91.tsd92.tsd93.tsd94.tsd95.tsd96.tsd97.tsd98.tsd99,
tsd100.tsd101.tsd102.tsd103.tsd104.tsd105.tsd106.tsd107.tsd108,

```

```

tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127,tsd128)
as select
  l_shipdate      ,
  l_orderkey      ,
  l_discount      ,
  l_extendedprice ,
  l_supplykey     ,
  l_quantity      ,
  l_returnflag    ,
  l_partkey       ,
  l_linestatus    ,
  l_tax           ,
  l_commitdate    ,
  l_receiptdate   ,
  l_shipmode      ,
  l_linenumbr     ,
  l_shipinstruct  ,
  l_comment
from l_et_$(i)
order by l_orderkey
;
commit;

lecho end loading table lineitem_items$(i)
!date
!!
----- dapopo_AUDIT.sh -----
#!/bin/ksh
i=$1
sqlplus /NOLOG <<!!

connect tpch/tpch
drop table o_et$(i);
create table o_et$(i)(
  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 fl1
access parameters
(
  records delimited by newline
  nobadfile
  nologfile
  fields terminated by '|'
  missing field values are null
)
)
location (
ff1:'orders.tbl_$(i)', ff2:'orders.tbl_$(i)', ff3:'orders.tbl_$(i)', ff4:'orders.tbl_$(i)',
ff5:'orders.tbl_$(i)', ff6:'orders.tbl_$(i)', ff7:'orders.tbl_$(i)', ff8:'orders.tbl_$(i)',
ff9:'orders.tbl_$(i)', ff10:'orders.tbl_$(i)', ff11:'orders.tbl_$(i)', ff12:'orders.tbl_$(i)',
ff13:'orders.tbl_$(i)', ff14:'orders.tbl_$(i)', ff15:'orders.tbl_$(i)', ff16:'orders.tbl_$(i)',
ff17:'orders.tbl_$(i)', ff18:'orders.tbl_$(i)', ff19:'orders.tbl_$(i)', ff20:'orders.tbl_$(i)',
ff21:'orders.tbl_$(i)', ff22:'orders.tbl_$(i)', ff23:'orders.tbl_$(i)', ff24:'orders.tbl_$(i)',
ff25:'orders.tbl_$(i)', ff26:'orders.tbl_$(i)', ff27:'orders.tbl_$(i)', ff28:'orders.tbl_$(i)',
ff29:'orders.tbl_$(i)', ff30:'orders.tbl_$(i)', ff31:'orders.tbl_$(i)', ff32:'orders.tbl_$(i)',
ff33:'orders.tbl_$(i)', ff34:'orders.tbl_$(i)', ff35:'orders.tbl_$(i)', ff36:'orders.tbl_$(i)',
ff37:'orders.tbl_$(i)', ff38:'orders.tbl_$(i)', ff39:'orders.tbl_$(i)', ff40:'orders.tbl_$(i)',
ff41:'orders.tbl_$(i)', ff42:'orders.tbl_$(i)', ff43:'orders.tbl_$(i)', ff44:'orders.tbl_$(i)',
ff45:'orders.tbl_$(i)', ff46:'orders.tbl_$(i)', ff47:'orders.tbl_$(i)', ff48:'orders.tbl_$(i)',
ff49:'orders.tbl_$(i)', ff50:'orders.tbl_$(i)', ff51:'orders.tbl_$(i)', ff52:'orders.tbl_$(i)',
ff53:'orders.tbl_$(i)', ff54:'orders.tbl_$(i)', ff55:'orders.tbl_$(i)', ff56:'orders.tbl_$(i)',
ff57:'orders.tbl_$(i)', ff58:'orders.tbl_$(i)', ff59:'orders.tbl_$(i)', ff60:'orders.tbl_$(i)',
ff61:'orders.tbl_$(i)', ff62:'orders.tbl_$(i)', ff63:'orders.tbl_$(i)', ff64:'orders.tbl_$(i)',
ff65:'orders.tbl_$(i)', ff66:'orders.tbl_$(i)', ff67:'orders.tbl_$(i)', ff68:'orders.tbl_$(i)',
ff69:'orders.tbl_$(i)', ff70:'orders.tbl_$(i)', ff71:'orders.tbl_$(i)', ff72:'orders.tbl_$(i)',
ff73:'orders.tbl_$(i)', ff74:'orders.tbl_$(i)', ff75:'orders.tbl_$(i)', ff76:'orders.tbl_$(i)',
ff77:'orders.tbl_$(i)', ff78:'orders.tbl_$(i)', ff79:'orders.tbl_$(i)', ff80:'orders.tbl_$(i)',
ff81:'orders.tbl_$(i)', ff82:'orders.tbl_$(i)', ff83:'orders.tbl_$(i)', ff84:'orders.tbl_$(i)',
ff85:'orders.tbl_$(i)', ff86:'orders.tbl_$(i)', ff87:'orders.tbl_$(i)', ff88:'orders.tbl_$(i)',
ff89:'orders.tbl_$(i)', ff90:'orders.tbl_$(i)', ff91:'orders.tbl_$(i)', ff92:'orders.tbl_$(i)',
ff93:'orders.tbl_$(i)', ff94:'orders.tbl_$(i)', ff95:'orders.tbl_$(i)', ff96:'orders.tbl_$(i)',
ff97:'orders.tbl_$(i)', ff98:'orders.tbl_$(i)', ff99:'orders.tbl_$(i)', ff100:'orders.tbl_$(i)',
ff101:'orders.tbl_$(i)', ff102:'orders.tbl_$(i)', ff103:'orders.tbl_$(i)', ff104:'orders.tbl_$(i)',
ff105:'orders.tbl_$(i)', ff106:'orders.tbl_$(i)', ff107:'orders.tbl_$(i)', ff108:'orders.tbl_$(i)',
ff109:'orders.tbl_$(i)', ff110:'orders.tbl_$(i)', ff111:'orders.tbl_$(i)', ff112:'orders.tbl_$(i)',
ff113:'orders.tbl_$(i)', ff114:'orders.tbl_$(i)', ff115:'orders.tbl_$(i)', ff116:'orders.tbl_$(i)',
ff117:'orders.tbl_$(i)', ff118:'orders.tbl_$(i)', ff119:'orders.tbl_$(i)', ff120:'orders.tbl_$(i)',
ff121:'orders.tbl_$(i)', ff122:'orders.tbl_$(i)', ff123:'orders.tbl_$(i)', ff124:'orders.tbl_$(i)',
ff125:'orders.tbl_$(i)', ff126:'orders.tbl_$(i)', ff127:'orders.tbl_$(i)', ff128:'orders.tbl_$(i)'
);reject limit unlimited;

alter table o_et$(i) parallel 4;

```

```

alter session force parallel dml parallel (degree 4);
set timing on
set echo on

lecho start loading table orders_ord$(i)
!date

drop table orders_ord$(i);
create table orders_ord$(i)(
  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
intrans 10
storage (initial 1m next 16m pctincrease 0 freelists groups 8 freelists 99)
compress
parallel 4
nologging
partition by hash(o_custkey)
partitions 256
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127,tsd128)
as select
  o_orderdate      ,
  o_orderkey        ,
  o_custkey         ,
  o_orderpriority   ,
  o_shippriority    ,
  o_clerk           ,
  o_orderstatus     ,
  o_totalprice      ,
  o_comment
from o_et$(i)
order by o_orderkey;
commit;

lecho end loading table orders_ord$(i)
!date
!!
----- dapopre_AUDIT.sh -----
#!/bin/bash
sqlplus /NOLOG <<EOF
connect tpch/tpch

drop table ps_et;
create table ps_et(
  ps_partkey       number ,
  ps_supplykey     number ,
  ps_availqty      number ,
  ps_supplycost    number ,
  ps_comment        varchar(199)
)
organization external (
type ORACLE_LOADER
default directory fl1
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',ff33:'partsupp.tbl.33',
ff34:'partsupp.tbl.34',ff35:'partsupp.tbl.35',ff36:'partsupp.tbl.36',
ff37:'partsupp.tbl.37',ff38:'partsupp.tbl.38',ff39:'partsupp.tbl.39',
ff40:'partsupp.tbl.40',ff41:'partsupp.tbl.41',ff42:'partsupp.tbl.42',
ff43:'partsupp.tbl.43',ff44:'partsupp.tbl.44',ff45:'partsupp.tbl.45',
ff46:'partsupp.tbl.46',ff47:'partsupp.tbl.47',ff48:'partsupp.tbl.48',
ff49:'partsupp.tbl.49',ff50:'partsupp.tbl.50',ff51:'partsupp.tbl.51',
ff52:'partsupp.tbl.52',ff53:'partsupp.tbl.53',ff54:'partsupp.tbl.54',
ff55:'partsupp.tbl.55',ff56:'partsupp.tbl.56',ff57:'partsupp.tbl.57',
ff58:'partsupp.tbl.58',ff59:'partsupp.tbl.59',ff60:'partsupp.tbl.60',
ff61:'partsupp.tbl.61',ff62:'partsupp.tbl.62',ff63:'partsupp.tbl.63',
ff64:'partsupp.tbl.64',ff65:'partsupp.tbl.65',ff66:'partsupp.tbl.66',
ff67:'partsupp.tbl.67',ff68:'partsupp.tbl.68',
ff69:'partsupp.tbl.69',ff70:'partsupp.tbl.70',
ff71:'partsupp.tbl.71',ff72:'partsupp.tbl.72',
ff73:'partsupp.tbl.73',ff74:'partsupp.tbl.74',ff75:'partsupp.tbl.75',
ff76:'partsupp.tbl.76',ff77:'partsupp.tbl.77',ff78:'partsupp.tbl.78',
ff79:'partsupp.tbl.79',ff80:'partsupp.tbl.80',ff81:'partsupp.tbl.81',
ff82:'partsupp.tbl.82',ff83:'partsupp.tbl.83',ff84:'partsupp.tbl.84',
ff85:'partsupp.tbl.85',ff86:'partsupp.tbl.86',ff87:'partsupp.tbl.87',
ff88:'partsupp.tbl.88',ff89:'partsupp.tbl.89',ff90:'partsupp.tbl.90',
ff91:'partsupp.tbl.91',ff92:'partsupp.tbl.92',ff93:'partsupp.tbl.93',
ff94:'partsupp.tbl.94',ff95:'partsupp.tbl.95',ff96:'partsupp.tbl.96',
ff97:'partsupp.tbl.97',ff98:'partsupp.tbl.98',ff99:'partsupp.tbl.99',
ff100:'partsupp.tbl.100',ff101:'partsupp.tbl.101',ff102:'partsupp.tbl.102',
ff103:'partsupp.tbl.103',ff104:'partsupp.tbl.104',ff105:'partsupp.tbl.105',
ff106:'partsupp.tbl.106',ff107:'partsupp.tbl.107',ff108:'partsupp.tbl.108',
ff109:'partsupp.tbl.109',ff110:'partsupp.tbl.110',ff111:'partsupp.tbl.111',
ff112:'partsupp.tbl.112',ff113:'partsupp.tbl.113',ff114:'partsupp.tbl.114',
ff115:'partsupp.tbl.115',ff116:'partsupp.tbl.116',ff117:'partsupp.tbl.117',
ff118:'partsupp.tbl.118',ff119:'partsupp.tbl.119',ff120:'partsupp.tbl.120',
ff121:'partsupp.tbl.121',ff122:'partsupp.tbl.122',ff123:'partsupp.tbl.123',
ff124:'partsupp.tbl.124',ff125:'partsupp.tbl.125',ff126:'partsupp.tbl.126',
ff127:'partsupp.tbl.127',ff128:'partsupp.tbl.128'
)
)
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',ff33:'part.tbl.33',
ff34:'part.tbl.34',ff35:'part.tbl.35',ff36:'part.tbl.36',
ff37:'part.tbl.37',ff38:'part.tbl.38',ff39:'part.tbl.39',
ff40:'part.tbl.40',ff41:'part.tbl.41',ff42:'part.tbl.42',
ff43:'part.tbl.43',ff44:'part.tbl.44',ff45:'part.tbl.45',
ff46:'part.tbl.46',ff47:'part.tbl.47',ff48:'part.tbl.48',
ff49:'part.tbl.49',ff50:'part.tbl.50',ff51:'part.tbl.51',
ff52:'part.tbl.52',ff53:'part.tbl.53',ff54:'part.tbl.54',
ff55:'part.tbl.55',ff56:'part.tbl.56',ff57:'part.tbl.57',
ff58:'part.tbl.58',ff59:'part.tbl.59',ff60:'part.tbl.60',
ff61:'part.tbl.61',ff62:'part.tbl.62',ff63:'part.tbl.63',
ff64:'part.tbl.64',ff65:'part.tbl.65',ff66:'part.tbl.66',
ff67:'part.tbl.67',ff68:'part.tbl.68',
ff69:'part.tbl.69',ff70:'part.tbl.70',
ff71:'part.tbl.71',ff72:'part.tbl.72',
ff73:'part.tbl.73',ff74:'part.tbl.74',ff75:'part.tbl.75',
ff76:'part.tbl.76',ff77:'part.tbl.77',ff78:'part.tbl.78',
ff79:'part.tbl.79',ff80:'part.tbl.80',ff81:'part.tbl.81',
ff82:'part.tbl.82',ff83:'part.tbl.83',ff84:'part.tbl.84',
ff85:'part.tbl.85',ff86:'part.tbl.86',ff87:'part.tbl.87',
ff88:'part.tbl.88',ff89:'part.tbl.89',ff90:'part.tbl.90',

```

```

ff91:'part.tbl.91',ff92:'part.tbl.92',ff93:'part.tbl.93',
ff94:'part.tbl.94',ff95:'part.tbl.95',ff96:'part.tbl.96',
ff97:'part.tbl.97',ff98:'part.tbl.98',ff99:'part.tbl.99',
ff100:'part.tbl.100',ff101:'part.tbl.101',ff102:'part.tbl.102',
ff103:'part.tbl.103',ff104:'part.tbl.104',ff105:'part.tbl.105',
ff106:'part.tbl.106',ff107:'part.tbl.107',ff108:'part.tbl.108',
ff109:'part.tbl.109',ff110:'part.tbl.110',ff111:'part.tbl.111',
ff112:'part.tbl.112',ff113:'part.tbl.113',ff114:'part.tbl.114',
ff115:'part.tbl.115',ff116:'part.tbl.116',ff117:'part.tbl.117',
ff118:'part.tbl.118',ff119:'part.tbl.119',ff120:'part.tbl.120',
ff121:'part.tbl.121',ff122:'part.tbl.122',ff123:'part.tbl.123',
ff124:'part.tbl.124',ff125:'part.tbl.125',ff126:'part.tbl.126',
ff127:'part.tbl.127',ff128:'part.tbl.128'
)
)
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',ff33:'customer.tbl.33',
ff34:'customer.tbl.34',ff35:'customer.tbl.35',ff36:'customer.tbl.36',
ff37:'customer.tbl.37',ff38:'customer.tbl.38',ff39:'customer.tbl.39',
ff40:'customer.tbl.40',ff41:'customer.tbl.41',ff42:'customer.tbl.42',
ff43:'customer.tbl.43',ff44:'customer.tbl.44',ff45:'customer.tbl.45',
ff46:'customer.tbl.46',ff47:'customer.tbl.47',ff48:'customer.tbl.48',
ff49:'customer.tbl.49',ff50:'customer.tbl.50',ff51:'customer.tbl.51',
ff52:'customer.tbl.52',ff53:'customer.tbl.53',ff54:'customer.tbl.54',
ff55:'customer.tbl.55',ff56:'customer.tbl.56',ff57:'customer.tbl.57',
ff58:'customer.tbl.58',ff59:'customer.tbl.59',ff60:'customer.tbl.60',
ff61:'customer.tbl.61',ff62:'customer.tbl.62',ff63:'customer.tbl.63',
ff64:'customer.tbl.64',ff65:'customer.tbl.65',ff66:'customer.tbl.66',
ff67:'customer.tbl.67',ff68:'customer.tbl.68',
ff69:'customer.tbl.69',ff70:'customer.tbl.70',
ff71:'customer.tbl.71',ff72:'customer.tbl.72',
ff73:'customer.tbl.73',ff74:'customer.tbl.74',ff75:'customer.tbl.75',
ff76:'customer.tbl.76',ff77:'customer.tbl.77',ff78:'customer.tbl.78',
ff79:'customer.tbl.79',ff80:'customer.tbl.80',ff81:'customer.tbl.81',
ff82:'customer.tbl.82',ff83:'customer.tbl.83',ff84:'customer.tbl.84',
ff85:'customer.tbl.85',ff86:'customer.tbl.86',ff87:'customer.tbl.87',
ff88:'customer.tbl.88',ff89:'customer.tbl.89',ff90:'customer.tbl.90',
ff91:'customer.tbl.91',ff92:'customer.tbl.92',ff93:'customer.tbl.93',
ff94:'customer.tbl.94',ff95:'customer.tbl.95',ff96:'customer.tbl.96',
ff97:'customer.tbl.97',ff98:'customer.tbl.98',ff99:'customer.tbl.99',
ff100:'customer.tbl.100',ff101:'customer.tbl.101',ff102:'customer.tbl.102',
ff103:'customer.tbl.103',ff104:'customer.tbl.104',ff105:'customer.tbl.105',
ff106:'customer.tbl.106',ff107:'customer.tbl.107',ff108:'customer.tbl.108',
ff109:'customer.tbl.109',ff110:'customer.tbl.110',ff111:'customer.tbl.111',
ff112:'customer.tbl.112',ff113:'customer.tbl.113',ff114:'customer.tbl.114',
ff115:'customer.tbl.115',ff116:'customer.tbl.116',ff117:'customer.tbl.117',
ff118:'customer.tbl.118',ff119:'customer.tbl.119',ff120:'customer.tbl.120',
ff121:'customer.tbl.121',ff122:'customer.tbl.122',ff123:'customer.tbl.123',
ff124:'customer.tbl.124',ff125:'customer.tbl.125',ff126:'customer.tbl.126',
ff127:'customer.tbl.127',ff128:'customer.tbl.128'
)
)
reject limit unlimited;

drop table s_et;
create table s_et(
  s_supplykey   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
nbadfile
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',ff33:'supplier.tbl.33',
ff34:'supplier.tbl.34',ff35:'supplier.tbl.35',ff36:'supplier.tbl.36',
ff37:'supplier.tbl.37',ff38:'supplier.tbl.38',ff39:'supplier.tbl.39',
ff40:'supplier.tbl.40',ff41:'supplier.tbl.41',ff42:'supplier.tbl.42',
ff43:'supplier.tbl.43',ff44:'supplier.tbl.44',ff45:'supplier.tbl.45',
ff46:'supplier.tbl.46',ff47:'supplier.tbl.47',ff48:'supplier.tbl.48',
ff49:'supplier.tbl.49',ff50:'supplier.tbl.50',ff51:'supplier.tbl.51',
ff52:'supplier.tbl.52',ff53:'supplier.tbl.53',ff54:'supplier.tbl.54',
ff55:'supplier.tbl.55',ff56:'supplier.tbl.56',ff57:'supplier.tbl.57',
ff58:'supplier.tbl.58',ff59:'supplier.tbl.59',ff60:'supplier.tbl.60',
ff61:'supplier.tbl.61',ff62:'supplier.tbl.62',ff63:'supplier.tbl.63',
ff64:'supplier.tbl.64',ff65:'supplier.tbl.65',ff66:'supplier.tbl.66',
ff67:'supplier.tbl.67',ff68:'supplier.tbl.68',
ff69:'supplier.tbl.69',ff70:'supplier.tbl.70',
ff71:'supplier.tbl.71',ff72:'supplier.tbl.72',
ff73:'supplier.tbl.73',ff74:'supplier.tbl.74',ff75:'supplier.tbl.75',
ff76:'supplier.tbl.76',ff77:'supplier.tbl.77',ff78:'supplier.tbl.78',
ff79:'supplier.tbl.79',ff80:'supplier.tbl.80',ff81:'supplier.tbl.81',
ff82:'supplier.tbl.82',ff83:'supplier.tbl.83',ff84:'supplier.tbl.84',
ff85:'supplier.tbl.85',ff86:'supplier.tbl.86',ff87:'supplier.tbl.87',
ff88:'supplier.tbl.88',ff89:'supplier.tbl.89',ff90:'supplier.tbl.90',
ff91:'supplier.tbl.91',ff92:'supplier.tbl.92',ff93:'supplier.tbl.93',
ff94:'supplier.tbl.94',ff95:'supplier.tbl.95',ff96:'supplier.tbl.96',
ff97:'supplier.tbl.97',ff98:'supplier.tbl.98',ff99:'supplier.tbl.99',
ff100:'supplier.tbl.100',ff101:'supplier.tbl.101',ff102:'supplier.tbl.102',
ff103:'supplier.tbl.103',ff104:'supplier.tbl.104',ff105:'supplier.tbl.105',
ff106:'supplier.tbl.106',ff107:'supplier.tbl.107',ff108:'supplier.tbl.108',
ff109:'supplier.tbl.109',ff110:'supplier.tbl.110',ff111:'supplier.tbl.111',
ff112:'supplier.tbl.112',ff113:'supplier.tbl.113',ff114:'supplier.tbl.114',
ff115:'supplier.tbl.115',ff116:'supplier.tbl.116',ff117:'supplier.tbl.117',
ff118:'supplier.tbl.118',ff119:'supplier.tbl.119',ff120:'supplier.tbl.120',
ff121:'supplier.tbl.121',ff122:'supplier.tbl.122',ff123:'supplier.tbl.123',
ff124:'supplier.tbl.124',ff125:'supplier.tbl.125',ff126:'supplier.tbl.126',
ff127:'supplier.tbl.127',ff128:'supplier.tbl.128'
))
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
nbadfile
nologfile
fields terminated by '|'
missing field values are null
)
)
location (
ff1:'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
nbadfile
nologfile
fields terminated by '|'
missing field values are null
)
)
location (
ff1:'region.tbl'
)
)
reject limit unlimited;

alter table l_et parallel 64;
alter table o_et parallel 64;
alter table ps_et parallel 64;
alter table p_et parallel 64;
alter table c_et parallel 64;
alter table s_et parallel 64;

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 partsupp;
create table partsupp(
ps_partkey      NOT NULL,
ps_suppkey      NOT NULL,
ps_supplycost   NOT NULL,
ps_availqty     ,
ps_comment      )
)
storage (initial 50m freelist groups 8 freelists 99)
parallel 64
nologging
partition by hash(ps_partkey)
partitions 256
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128)
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 50m freelist groups 8 freelists 99)
compress
parallel 64
nologging
partition by hash(c_custkey)
partitions 256
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,

```

```

tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127,tsd128)
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_mfg         ,
  p_retailprice ,
  p_comment
)
pctfree 0
pctused 99
storage (initial 66m freelist groups 8 freelists 99)
compress
parallel 64
nologging
partition by hash (p_partkey)
partitions 256
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127,tsd128)
as select
  p_partkey      ,
  p_type         ,
  p_size         ,
  p_brand        ,
  p_name         ,
  p_container    ,
  p_mfg         ,
  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 80m freelist groups 8 freelists 99)
compress
parallel 64
nologging
partition by hash (s_suppkey)
partitions 256
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127,tsd128)

```

```

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

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 256
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127,tsd128)
pctfree 2
intrans 10
storage (initial 60m freelist groups 8 freelists 99)
parallel 64
compute statistics
nologging;
!date

rem drop index i_o_orderkey;
create unique index i_o_orderkey
on orders (o_orderkey) global partition by hash (o_orderkey)
partitions 256
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127,tsd128)
pctfree 2
intrans 10
storage (initial 80m freelist groups 8 freelists 99)
parallel 64
compute statistics
nologging;
!date

rem drop index i_c_custkey;
create unique index i_c_custkey
on customer (c_custkey) global partition by hash (c_custkey)
partitions 256

```

```

store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127,tsd128)
pctfree 2
intrans 10
storage (initial 50m freelist groups 8 freelists 99)
parallel 64
compute statistics
nologging;

create unique index i_ps_partkey_supkey
on partsupp (ps_partkey,ps_supkey) global partition by hash (ps_partkey)
partitions 256
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127,tsd128)
pctfree 2
intrans 10
storage (initial 64m freelist groups 8 freelists 99)
parallel 64
compute statistics
nologging;

alter index i_l_orderkey allocate extent (size 660m);
alter index i_l_orderkey allocate extent (size 660m);
alter index i_l_orderkey allocate extent (size 660m);
alter index i_l_orderkey allocate extent (size 660m);

alter index i_o_orderkey allocate extent (size 80m);
alter index i_o_orderkey allocate extent (size 80m);
alter index i_o_orderkey allocate extent (size 80m);
alter index i_o_orderkey allocate extent (size 80m);

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

sqlplus /NOLOG <<!
connect /as sysdba
shutdown abort
startup pfile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora nomount;
create database
controlfile reuse
logfile '/home/oracle/dev/block/log_1' size 1024m reuse,
'/home/oracle/dev/block/log_2' size 1024m reuse,
'/home/oracle/dev/block/log_1_a' size 1024m reuse,
'/home/oracle/dev/block/log_2_a' size 1024m reuse
datafile '/home/oracle/dev/block/sys1' size 512m reuse,
'/home/oracle/dev/block/sys2' size 512m reuse,
'/home/oracle/dev/block/sys3' size 512m reuse,
'/home/oracle/dev/block/sys4' size 512m reuse,
'/home/oracle/dev/block/sys5' size 512m reuse,
'/home/oracle/dev/block/sys6' size 512m reuse,
'/home/oracle/dev/block/sys7' size 512m reuse,
'/home/oracle/dev/block/sys8' size 512m reuse
sysaux datafile '/home/oracle/dev/block/sysaux_1' size 4096m reuse,
'/home/oracle/dev/block/sysaux_2' size 4096m reuse,
'/home/oracle/dev/block/sysaux_3' size 4096m reuse
undo tablespace ts_undo1 datafile '/home/oracle/dev/block/undo_1' size 1536m reuse
default temporary tablespace ts_temp
tempfile '/home/oracle/dev/block/t_1' size 27000m reuse
extent management local uniform size 10m
maxlogfiles 128
maxdatafiles 4000
maxinstances 64;

set termout off
set echo off
spool /tmp/cat
@?/rdbs/admin/catalog.sql;

```

```

@?/rdbs/admin/catproc.sql;
@?/rdbs/admin/catclust.sql;
connect system/manager
@?/sqlplus/admin/publbd.sql;
spool off
!

(( n=1 ))
while (( n<8 ));do
(( n=n+1 ))
addundolog.sh $n
done
wait

tshut

for i in 101 102 103 104 105 106 107;do
ssh high5{i} -n /home/oracle/frame/bin/tstart
done

for i in 101 102 103 104 105 106 107;do
ssh high5{i} -n /home/oracle/kit/schema/10.0/build/asm_db/add_remaining_undolog.sh &
done
wait
2start
wait
echo end database creation `date`
----- drop_temp_part.sh -----

n=`count.sh $1 | tail -4 | head -1`
if ((n>0)); then
echo nothing to be done!
else
sqlplus /NOLOG <<!!
connect tpch/tpch
drop table $1;
--select count(*) from $1;
!!
fi
----- ex_linparts.sh -----
#!/bin/ksh

sqlplus /NOLOG << !!
connect tpch/tpch
drop table lineitem;
create table lineitem(
l_shipdate date,
l_orderkey number NOT NULL,
l_discount number NOT NULL,
l_extendedprice number NOT NULL,
l_suppkey number NOT NULL,
l_quantity number NOT NULL,
l_returnflag char(1),
l_partkey number NOT NULL,
l_linestatus char(1),
l_tax number NOT NULL,
l_commitdate date,
l_receiptdate date,
l_shipmode char(10),
l_linenum number NOT NULL,
l_shipinstruct char(25),
l_comment varchar(44)
)
pctfree 1
pctused 99
intrans 10
storage (initial 1m next 16m pctincrease 0 freelist groups 8 freelists 99)
compress
parallel 256
nologging
partition by range (l_shipdate)
subpartition by hash(l_partkey)
subpartitions 256
(
partition item1 values less than (to_date('1992-01-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127,tsd128),
partition item2 values less than (to_date('1992-02-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,

```



```

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}/rdtablest
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
wait
STIME=$GTIME
SLTIME=$GTIME
echo "Start: timed load portion `date`" >> $SCRIPT_LOG_FILE
/home/oracle/kit/audit/dapop_AUDIT.sh >> $LD3DAPOP
$KIT_DIR/audit/gen_seed.sh $KIT_DIR/audit/seed
echo "End: timed load portion `date`" >> $SCRIPT_LOG_FILE
echo Generated seed: `cat $KIT_DIR/audit/seed` >> $SCRIPT_LOG_FILE
----- tscrc_AUDIT.sh -----
#!/bin/ksh
echo "START: tablespace creation"
date;

```

```

sqlplus /NOLOG <<!
connect /as sysdba
drop tablespace ts_default including contents;
create tablespace ts_default
datafile '/home/oracle/dev/block/default1' size 512m reuse
extent management dictionary default storage (initial 10m next 10m maxextents unlimited
pctincrease 0);
alter tablespace ts_default add datafile '/home/oracle/dev/block/default2' size 512m reuse;
!
wait;

i=0
while [ $i -lt 64 ]
#while [ $i -lt 128 ]
do
  i=`expr $i + 1`
  (( node=100+i ))
  (( first=(i-1)*2+1 ))
  (( second=first+1 ))
  ssh bigh${node} /home/oracle/kit/audit/crts.sh $first 38000m &
  ssh bigh${node} /home/oracle/kit/audit/crts.sh $second 38000m &
done
wait;
exit;
i=1
while [ $i -lt 128 ]
do
  i=`expr $i + 1`
  /home/oracle/kit/audit/addtts.sh ts_temp /home/oracle/dev/block/t_${i} 32000m &
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_supkey
Rem The range for ps_partkey is 1 to 20000
Rem The range for ps_supkey is 1 to 1000
Rem A valid combination is 46 and 47
Rem Usage: sqlplus tpcd/tpcd @a_query2 <ps_partkey> <ps_supkey>
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_supkey
rem The range for ps_partkey is 1 to 20000
rem The range for ps_supkey 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_supkey = &&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
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
#

. SKIT_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 atransplott"
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=$KIT_DIR/utls/atranspl
OUT=${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 ""

SKIT_DIR/utls/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 ""

SKIT_DIR/utls/andkey $ITER $$F 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 brand48();

/* 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 *tpscvc = 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> [<pathname for output>] [<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");
    fprintf(stderr, "delta :1 to generate new random delta, otherwise obtain delta from
input\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/tcpd\n");
    fprintf(stderr, "t-<trigger> :Trigger Time - sleep <trigger> seconds before start\n");
    fprintf(stderr, "s<sleep> :Sleep Time - sleep <sleep> seconds before commit or
rollback\n");
    exit(-1);
}

void ACIDexit() {

```

```

OCILogoff(tpcsvc,errhp);
OCIhfree(tpcenv,OCI_HTYPE_STMT);
OCIhfree(tpcsrc,OCI_HTYPE_SVCCCTX);
OCIhfree(tpcsrc,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(tpcsrc,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':
strcpy((char *) lname, ++(argv[0]), UNAME_LEN);
if (strcmp((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
scanf(line, "%d %d\n", &co_key, &delta);

```

```

/* Obtain L_key from L_key query */
OCIExec(tpcsvc,curi,errhp,1);

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

L_key = (int) ((rand48() % L_key) + 1);
#else
scanf(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);

OCIExec(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_епrice);
    FPRTF1(outfile, "L_quantity: %d\n", (int) L_quan);
    FPRTF1(outfile, "o_totalprice: %.2f\n", o_tprice);
}

FPRTF1(outfile, "Sleep %d seconds before COMMIT/ROLLBACK...\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);
            OCIExec(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));
    FPRTF1(logfile, "-----\n");
} else {

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

    FPRTF(outfile, "AFTER TRANSACTION:\n");
    FPRTF1(outfile, "L_extendedprice: %.2f\n", L_newep);

```

```

FPRTF1(outfile, "L_quantity: %d\n", (int) L_newquan);
FPRTF1(outfile, "o_totalprice: %.2f\n", o_newtprice);
FPRTF1(outfile, "L_tax: %.2f\n", L_tax);
FPRTF1(outfile, "L_discount: %.2f\n", L_disc);
FPRTF1(outfile, "rprice: %.2f\n", rprice);
FPRTF1(outfile, "cost: %.2f\n", cost);
FPRTF(outfile, "-----\n");
}
}

tr_end = gettimeofday();

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);
    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,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),
              OCL_NTV_SYNTAX,OCL_DEFAULT);
OCIExec(tcpsvc,curi,errhp,1);

/* Enable session parallel ddl */

/*sprintf((char *) sqlstmt, PDDLTEXT);
OCIStmtPrepare(curi,errhp,(text *)sqlstmt,strlen((char *)sqlstmt),
              OCL_NTV_SYNTAX,OCL_DEFAULT);
OCIExec(tcpsvc,curi,errhp,1);*/

/* Make session serializable */

sprintf((char *) sqlstmt, ISOTXT);
OCIStmtPrepare(curi,errhp,(text *)sqlstmt,strlen((char *)sqlstmt),
              OCL_NTV_SYNTAX,OCL_DEFAULT);
OCIExec(tcpsvc,curi,errhp,1);

/* Set optimizer_index_cost_adj = 25 */

sprintf((char *) sqlstmt, OICATXT);
OCIStmtPrepare(curi,errhp,(text *)sqlstmt,strlen((char *)sqlstmt),
              OCL_NTV_SYNTAX,OCL_DEFAULT);
OCIExec(tcpsvc,curi,errhp,1);

curr_time = time(NULL);
printf("\nConnected to ORACLE as user: %s at %s\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),OCL_NTV_SYNTAX,OCL_DEFAULT);

OCIbname(curi,&l_key1_bp,errhp,":l_key",ADR(l_key),SIZ(l_key),SQLT_INT);
OCIbname(curi,&o_key1_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(curi,errhp,(text *)sqlstmt,strlen((char *)sqlstmt),
              OCL_NTV_SYNTAX,OCL_DEFAULT);

/* bind variables */

OCIbname(curi,l_key_bp,errhp,":l_key",ADR(l_key),SIZ(l_key),SQLT_INT);
OCIbname(curi,o_key_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),SQLT_INT);
OCIbname(curi,delta_bp,errhp,":delta",ADR(delta),SIZ(delta),SQLT_INT);
OCIbname(curi,l_pkey_bp,errhp,":l_pkey",ADR(l_pkey),SIZ(l_pkey),SQLT_INT);
OCIbname(curi,l_skey_bp,errhp,":l_skey",ADR(l_skey),SIZ(l_skey),SQLT_INT);
OCIbname(curi,l_quan_bp,errhp,":l_quan",ADR(l_quan),SIZ(l_quan),SQLT_INT);
OCIbname(curi,l_newquan_bp,errhp,":l_newquan",ADR(l_newquan),SIZ(l_newquan),SQLT_INT);
OCIbname(curi,l_tax_bp,errhp,":l_tax",ADR(l_tax),SIZ(l_tax),SQLT_FLT);
OCIbname(curi,l_disc_bp,errhp,":l_disc",ADR(l_disc),SIZ(l_disc),SQLT_FLT);
OCIbname(curi,l_eprice_bp,errhp,":l_eprice",ADR(l_eprice),SIZ(l_eprice),SQLT_FLT);
OCIbname(curi,l_newprice_bp,errhp,":l_newprice",ADR(l_newprice),SIZ(l_newprice),SQLT_FLT);

OCIbname(curi,o_tprice_bp,errhp,":o_tprice",ADR(o_tprice),SIZ(o_tprice),SQLT_FLT);
OCIbname(curi,o_newprice_bp,errhp,":o_newprice",ADR(o_newprice),SIZ(o_newprice),SQLT_FLT);
OCIbname(curi,rprice_bp,errhp,":rprice",ADR(rprice),SIZ(rprice),SQLT_FLT);
OCIbname(curi,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),
              OCL_NTV_SYNTAX,OCL_DEFAULT);

sprintf((char *) sqlstmt,SQLTXT4);
OCIStmtPrepare(cure2,errhp,(text *)sqlstmt,strlen((char *)sqlstmt),
              OCL_NTV_SYNTAX,OCL_DEFAULT);

/* bind variables */

OCIbname(cure1,l_newprice1_bp,errhp,":l_newprice",ADR(l_newprice),SIZ(l_newprice),SQLT_FLT);
OCIbname(cure1,l_newquan1_bp,errhp,":l_newquan",ADR(l_newquan),SIZ(l_newquan),SQLT_INT);

```

```

SIZ(l_newquan),SQLT_INT);
OCIbname(cure1,o_key1_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),SQLT_INT);
OCIbname(cure1,l_key1_bp,errhp,":l_key",ADR(l_key),SIZ(l_key),SQLT_INT);

OCIbname(cure2,o_newprice2_bp,errhp,":o_newprice",ADR(o_newprice),SIZ(o_newprice),SQLT_FLT);
OCIbname(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

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

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

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

extern int erro;

#ifdef NULL
#define NULL 0
#endif

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

#ifdef DISCARD
#define DISCARD (void)
#endif

#ifdef sword
#define sword int
#endif

#ifdef 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(flag,mask) (unsigned) (flag |= (unsigned) mask)
#define BIT(flag,mask) (unsigned) (flag & (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 OCIlget(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 OCIlaset(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=OCISmtExecute(svch,stmh,errh,iter,0,NULL,NULL,OCI_DEFAULT)) != OCI_SUCCESS) \
    sql_error(errh,status,1); \
else \
    DISCARD 0

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

#define OCIBbynamei(stmh,bindp,errh,sqlvar,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), \
    progvl,ftype,indp,0,0,0,0,OCI_DEFAULT)) != OCI_SUCCESS) \
    sql_error(errh,status,1); \
else \
    DISCARD 0

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

#define OCIcon(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 PDDLTX "alter session force parallel ddl parallel (degree 2)"
#define OICATXT "alter session set optimizer_index_cost_adj=10"

#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_newprice, \
    :o_tprice, :o_newtprice, :rprice, :cost); END;"

#define SQLTXT3 "BEGIN SELECT l_extendedprice, l_quantity \
    INTO :l_newprice, :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_newprice     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_newprice     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
#

.SKIT_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

```

```

#
.SKIT_DIR/env

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

KEY=$OUT_DIR/key$$
OUTFILE=${OUT_DIR}/constrte
CON1=${OUT_DIR}/comb
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 "-h             : 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 prvotopt 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
#

. SKIT_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/SHCNT 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 $DURAS{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 ("%0.2f\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
#

. SKIT_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
SPROG 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 >> $ISOFIILE

/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:

.SKIT_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
ISOFIILE=$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
SPROG 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 >> $ISOFIILE

/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
#
# =====+
# May need to change the following:

.SKIT_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=/home/oracle/kit/utills/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

.SKIT_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=/home/oracle/kit/utills/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$$
TXN2FILE=$OUT_DIR/txn2$$
KEYFILE=$OUT_DIR/key$$
ISOFILE=$OUT_DIR/iso5

USER=$DATABASE_USER
PROG=/home/oracle/kit/utills/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$$
TXN2FILE=$OUT_DIR/txn2$$
TXN3FILE=$OUT_DIR/txn3$$
KEYFILE=$OUT_DIR/key$$
ISOFILE=$OUT_DIR/iso6

USER=$DATABASE_USER
PROG=/home/oracle/kit/utills/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*15000*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_PARSE adopted from dss.h */

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

void sql_error();
void usage();
void ACIDinit();
long atol();
void srand48();
long brand48();

/* 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;
OCIError *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);
OCIfree(tpcenv,OCI_HTYPE_STMT);
OCIfree(tpcsvc,OCI_HTYPE_SVCCTX);
OCIfree(tpcsrv,OCI_HTYPE_SERVER);
OCIfree(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;

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");
flush(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 = atoi(argv[1]);
sf = atof(argv[2]);

argc -= 2;
argv += 2;

while (--argc) {
++argv;
switch(argv[0][0]) {
case 'u':
strcpy((char *) lname, ++(argv[0]), UNAME_LEN);
if (strcmp((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_PARSE((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_linenum 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(tpcenv,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(tpcenv, errhp, tpcusr, OCI_CRED_RDBMS,
OCI_DEFAULT)) != OCI_SUCCESS)
sql_error(errhp,status,1);

OCIaset(tpcenv,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_NT_V_SYNTAX,OCI_DEFAULT);

OCIbname(curi,l_key_bp,errhp,":l_key",ADR(l_key),SIZ(l_key),SQLT_INT);
OCIbname(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
#
. SKIT_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 atranspLott"

```

```

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
    SPROG $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
. SKIT_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: Qualification query text and output

```

Q1
Begin Execution at Tue Jun 6 16:59:41 2006

-- 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 9.07 seconds.

Ended Executing this Stream at Tue Jun 6 16:59:51 2006

Stream Started at 1149631181.97
Stream Ended at 1149631191.04
Stream Processed in 9.07 seconds

SQL statements processed: 1
Q2
Begin Execution at Tue Jun 6 16:59:51 2006

-- 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 across
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
B3rqp0xbSEim4Mpy2RHJ 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,E2fjVd7eUrzp2E8j1QxGo2DFnosaTEH 17-516-924-4574
final theodolites cajole slyly special,
9870.78 Supplier#000001286 GERMANY
181285.00 Manufacturer#4
YKA,E2fjVd7eUrzp2E8j1QxGo2DFnosaTEH 17-516-924-4574
final theodolites cajole slyly special,
9852.52 Supplier#000008973 RUSSIA
18972.00 Manufacturer#2
t5L67YdBYH6o,Vz24jpDyQ9 32-188-594-7038
quickly regular instructions wake-- carefully unusual braids into the expres
9847.83 Supplier#000008097 RUSSIA
130557.00 Manufacturer#2
xMe97bpE69NzdwLoX 32-375-640-3593
slyly regular dependencies sleep slyly furiously express dep
9847.57 Supplier#000006345 FRANCE
86344.00 Manufacturer#1
VSt3rzk3qG698u6ld8HhOByvrTcSTsvQIDQDag 16-886-766-7945
silent pinto beans should have to snooze carefully along the final reques
9847.57 Supplier#000006345 FRANCE
173827.00 Manufacturer#2
VSt3rzk3qG698u6ld8HhOByvrTcSTsvQIDQDag 16-886-766-7945

```

```

silent pinto beans should have to snooze carefully along the final reques
----- rows deleted -----
8328.46 Supplier#000001744 ROMANIA
69237.00 Manufacturer#5
oLo3fV64q2,FKHa3p,qHnS7Yzv,ps8 29-330-728-5873
blithely silent excuses are slyly above the furiously even courts
8307.93 Supplier#000003142 GERMANY
18139.00 Manufacturer#1
dqblvV8dCNaorGJl 17-595-447-6026
theodolites sleep blithely carefully regular warhorses. slyly regular ins
8231.61 Supplier#000009558 RUSSIA
192000.00 Manufacturer#2
mcdgen,yT1iJDHDS5fV 32-762-137-5858
slyly regular theodolites sleep fluffily express depos
8152.61 Supplier#000002731 ROMANIA
15227.00 Manufacturer#4
nluXJCuy1tu 29-805-463-2030
gifts use. slyly silent ideas are carefully beneath the silent instructions. slyly sil
8109.09 Supplier#000009186 FRANCE
99185.00 Manufacturer#1
wgfosrVPexl9pEXWywaqIBMDYYf 16-668-570-1402
quickly pending requests are blithely along the ironic, final requests; instr
8102.62 Supplier#000003347 UNITED KINGDOM
18344.00 Manufacturer#5
m CxS2S16i 33-454-274-8532
packages grow special orbits. regular theodolites about the carefully pe
8046.07 Supplier#000008780 FRANCE
191222.00 Manufacturer#3
AczzuE0UK9osj ,Lx0Jmh 16-473-215-6395
regular epitaphs integrate slyly.
8042.09 Supplier#000003245 RUSSIA
135705.00 Manufacturer#4
Dh8lkG39onrbOL4DyTfGw8a9oKUX3d9Y 32-836-132-8872
carefully regular instructions integrate blithely silent foxes. furiously express instructions hagg
8042.09 Supplier#000003245 RUSSIA
150729.00 Manufacturer#1
Dh8lkG39onrbOL4DyTfGw8a9oKUX3d9Y 32-836-132-8872
carefully regular instructions integrate blithely silent foxes. furiously express instructions hagg
7992.40 Supplier#000006108 FRANCE
118574.00 Manufacturer#1
8tBydnTDwUqfBfFV4l3 16-974-998-8937
regular pinto beans are after
7980.65 Supplier#000001288 FRANCE
13784.00 Manufacturer#4
zE,7HgVPrCn 16-646-464-8247
unusual pinto beans cajole furiously according t
7950.37 Supplier#000008101 GERMANY
33094.00 Manufacturer#5
kkYvL6lujovJgTNGlKkaXQDYgx8lLohj 17-627-663-8014
quickly regular requests are furiously. pending deposits wake
7937.93 Supplier#000009012 ROMANIA
83995.00 Manufacturer#2
iUfTziH,Ek3i4lwSgunXMgrcTzwdb 29-250-925-9690
blithely bold ideas haggle quickly final, regular request
7914.45 Supplier#000001013 RUSSIA
125988.00 Manufacturer#2
riRcntps4KEDtYScjpMIWeYf6mNnR 32-194-698-3365
final, ironic theodolites alongside of the ironic
7912.91 Supplier#000004211 GERMANY
159180.00 Manufacturer#5
2wQRVovHrm3,v03IKzTfD,1PYsFXQFFOG 17-266-947-7315
final requests integrate slyly above the silent, even
7912.91 Supplier#000004211 GERMANY
184210.00 Manufacturer#4
2wQRVovHrm3,v03IKzTfD,1PYsFXQFFOG 17-266-947-7315
final requests integrate slyly above the silent, even
7894.56 Supplier#000007981 GERMANY
85472.00 Manufacturer#4
NSJ96vMROAbeXP 17-963-404-3760
regular, even theodolites integrate carefully. bold, special theodolites are slyly fluffily iron
7887.08 Supplier#000009792 GERMANY
164759.00 Manufacturer#3
Y28ITVeYrT3kIGdV2K8fSZ V2UqT5H1Otz 17-988-938-4296
pending, ironic packages sleep among the carefully ironic accounts. quickly final accounts
7871.50 Supplier#000007206 RUSSIA
104695.00 Manufacturer#1
3w fNCnrVmvJJE95sgWZzvW 32-432-452-7731
furiously dogged pinto beans cajole. bold, express notornis until the slyly pending
7852.45 Supplier#000005864 RUSSIA
8363.00 Manufacturer#4
WCNfBPZeSXh3h,c 32-454-883-3821
blithely regular deposits
7850.66 Supplier#000001518 UNITED KINGDOM
86501.00 Manufacturer#1
ONda3YJiHKJOC 33-730-383-3892
furiously final accounts wake carefully idle requests. even dolphins wake acc
7843.52 Supplier#000006683 FRANCE
11680.00 Manufacturer#4
2Z0JGkiv01Y00oCFwUGfvilbhzCdy 16-464-517-8943
carefully bold accounts doub

```

100 rows processed.

Query Processed in 0.26 seconds.

Ended Executing this Stream at Tue Jun 6 16:59:51 2006

Stream Started at 1149631191.10
Stream Ended at 1149631191.36
Stream Processed in 0.26 seconds

SQL statements processed: 1
Q3
Begin Execution at Tue Jun 6 16:59:51 2006

-- 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 1.45 seconds.

Ended Executing this Stream at Tue Jun 6 16:59:52 2006

Stream Started at 1149631191.41
Stream Ended at 1149631192.87
Stream Processed in 1.45 seconds

SQL statements processed: 1
Q4
Begin Execution at Tue Jun 6 16:59:52 2006

-- 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 1.53 seconds.

Ended Executing this Stream at Tue Jun 6 16:59:54 2006

```
Stream Started at 1149631192.92
Stream Ended at 1149631194.45
Stream Processed in 1.53 seconds
```

```
SQL statements processed: 1
Q5
Begin Execution at Tue Jun 6 16:59:54 2006
```

-- 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 2.10 seconds.

Ended Executing this Stream at Tue Jun 6 16:59:56 2006

```
Stream Started at 1149631194.51
Stream Ended at 1149631196.61
Stream Processed in 2.10 seconds
```

```
SQL statements processed: 1
Q6
Begin Execution at Tue Jun 6 16:59:56 2006
```

-- 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 0.65 seconds.

Ended Executing this Stream at Tue Jun 6 16:59:57 2006

```
Stream Started at 1149631196.67
Stream Ended at 1149631197.32
Stream Processed in 0.65 seconds
```

```
SQL statements processed: 1
Q7
Begin Execution at Tue Jun 6 16:59:57 2006
```

-- 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 1.25 seconds.

Ended Executing this Stream at Tue Jun 6 16:59:58 2006

Stream Started at 1149631197.37
 Stream Ended at 1149631198.62
 Stream Processed in 1.25 seconds

SQL statements processed: 1
 Q8
 Begin Execution at Tue Jun 6 16:59:58 2006

-- 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 1.65 seconds.

Ended Executing this Stream at Tue Jun 6 17:00:00 2006

Stream Started at 1149631198.68
 Stream Ended at 1149631200.33
 Stream Processed in 1.65 seconds

SQL statements processed: 1
 Q9
 Begin Execution at Tue Jun 6 17:00:00 2006

-- 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,
partsupp,
orders,
lineitem,
```

```
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

```
----- rows deleted -----
UNITED KINGDOM 1998.00 28494874.00
UNITED KINGDOM 1997.00 49381810.90
UNITED KINGDOM 1996.00 51386853.96
UNITED KINGDOM 1995.00 51509586.79
UNITED KINGDOM 1994.00 48086499.71
UNITED KINGDOM 1993.00 49166827.22
UNITED KINGDOM 1992.00 49349122.08
UNITED STATES 1998.00 25126238.95
UNITED STATES 1997.00 50077306.42
UNITED STATES 1996.00 48048649.47
UNITED STATES 1995.00 48809032.42
UNITED STATES 1994.00 49296747.18
UNITED STATES 1993.00 48029946.80
UNITED STATES 1992.00 48671944.50
VIETNAM 1998.00 30442736.06
VIETNAM 1997.00 50309179.79
VIETNAM 1996.00 50488161.41
VIETNAM 1995.00 49658284.61
VIETNAM 1994.00 50596057.26
VIETNAM 1993.00 50953919.15
VIETNAM 1992.00 49613838.32
```

175 rows processed.
 Query Processed in 4.24 seconds.

Ended Executing this Stream at Tue Jun 6 17:00:04 2006

Stream Started at 1149631200.38
 Stream Ended at 1149631204.62
 Stream Processed in 4.24 seconds

SQL statements processed: 1
 Q10
 Begin Execution at Tue Jun 6 17:00:04 2006

-- 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
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	
IaReFYv.Kw4	14-742-935-3718	
fluffily bold excuses haggle finally after the u		
60838.00	Customer#000060838	679127.31
2454.77	BRAZIL	
64Eaj5vMAHWJIBOXJkIpNc2RJiWE	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 ne7zV5fDrmiqIoK09wV7pxqCglc	16-715-386-3788	
carefully bold excuses sleep alongside of the thinly idle		
84223.00	Customer#000084223	594998.02
528.65	UNITED KINGDOM	
nAVZCs6BaWap rrM27N 2qBnzc5WBauxbA	33-442-824-8191	
pending, final ideas haggle final requests. unusual, regular asymptotes affix according to the		
even foxes.		
54289.00	Customer#000054289	585603.39
5583.02	IRAN	
vXCxoCsU0Bad5JQI .oobkZ	20-834-292-4707	
express requests sublate blithely regular requests. regular, even ideas solve.		
39922.00	Customer#000039922	584878.11
7321.11	GERMANY	
Zgy4s50l2GKN4pLDPBU8m342glw6R	17-147-757-8036	
even pinto beans haggle. slyly bold accounts inte		
6226.00	Customer#000006226	576783.76
2230.09	UNITED KINGDOM	
8gPu8.NPGkfyQQ0hcIYUGPIBWc.ybP5g.	33-657-701-3391	
quickly final requests against the regular instructions wake blithely final instructions. pa		
922.00	Customer#00000922	576767.53
3869.25	GERMANY	
Az9RFaut7NkPnc5zSD2PwHgVwr4jrZq	17-945-916-9648	
boldly final requests cajole blith		
147946.00	Customer#000147946	576455.13
2030.13	ALGERIA	
iANyZHjqhy7Ajah0pTrYyhJ	10-886-956-3143	
furiously even accounts are blithely above the furiously		
115640.00	Customer#000115640	569341.19
6436.10	ARGENTINA	
Vtgfia9ql 7EpHgecU1X	11-411-543-4901	
final instructions are slyly according to the		
73606.00	Customer#000073606	568656.86
1785.67	JAPAN	
xuR0Tro5yChDfOCrjkd2ol	22-437-653-6966	
furiously bold orbits about the furiously busy requests wake across the furiously quiet		
theodolites. d		
110246.00	Customer#000110246	566842.98

```

7763.35 VIETNAM
7KzflgX MDOq7sOkI 31-943-426-9837
dolphins sleep blithely among the slyly final
142549.00 Customer#000142549 563537.24
5085.99 INDONESIA
ChqEoK43OysjdHbtKCP6dKqjNyvvi9 19-955-562-2398
regular, unusual dependencies boost slyly; ironic attainments nag fluffily into the unusual
packages?
146149.00 Customer#000146149 557254.99
1791.55 ROMANIA
s87fvzFQpU 29-744-164-6487
silent, unusual requests detect quickly slyly regul
52528.00 Customer#000052528 556397.35
551.79 ARGENTINA
NFztyTOR10UOJ 11-208-192-3205
unusual requests detect. slyly dogged theodolites use slyly. deposit
23431.00 Customer#000023431 554269.54
3381.86 ROMANIA
HgiV0pqhala9aydNoIlb 29-915-458-2654
instructions nag quickly. furiously bold accounts cajol

```

20 rows processed.
Query Processed in 2.06 seconds.

Ended Executing this Stream at Tue Jun 6 17:00:06 2006

Stream Started at 1149631204.68
Stream Ended at 1149631206.74
Stream Processed in 2.06 seconds

SQL statements processed: 1
Q11
Begin Execution at Tue Jun 6 17:00:06 2006

-- 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
-----rows deleted-----	
72922.00	7930400.64
146699.00	7929167.40
92387.00	7928972.67
186289.00	7928786.19
95952.00	7927972.78

196514.00 7927180.70
 4403.00 7925729.04
 2267.00 7925649.37
 45924.00 7925047.68
 11493.00 7916722.23
 104478.00 7916253.60
 166794.00 7913842.00
 161995.00 7910874.27
 23538.00 7909752.06
 41093.00 7909579.92
 112073.00 7908617.57
 92814.00 7908262.50
 88919.00 7907992.50
 79753.00 7907933.88
 108765.00 7905338.98
 146530.00 7905336.60
 71475.00 7903367.58
 36289.00 7901946.50
 61739.00 7900794.00
 52338.00 7898638.08
 194299.00 7898421.24
 105235.00 7897829.94
 77207.00 7897752.72
 96712.00 7897575.27
 10157.00 7897046.25
 171154.00 7896814.50
 79373.00 7896186.00
 113808.00 7893353.88
 27901.00 7892952.00
 128820.00 7892882.72
 25891.00 7890511.20
 122819.00 7888881.02
 154731.00 7888301.33
 101674.00 7879324.60
 51968.00 7879102.21
 72073.00 7877736.11
 5182.00 7874521.73

1048 rows processed.
 Query Processed in 0.42 seconds.

Ended Executing this Stream at Tue Jun 6 17:00:07 2006

Stream Started at 1149631206.79
 Stream Ended at 1149631207.22
 Stream Processed in 0.42 seconds

SQL statements processed: 1
 Q12
 Begin Execution at Tue Jun 6 17:00:07 2006

-- 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 1.12 seconds.

Ended Executing this Stream at Tue Jun 6 17:00:08 2006

Stream Started at 1149631207.27
 Stream Ended at 1149631208.39
 Stream Processed in 1.12 seconds

SQL statements processed: 1
 Q13
 Begin Execution at Tue Jun 6 17:00:08 2006

-- 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 3.17 seconds.

Ended Executing this Stream at Tue Jun 6 17:00:11 2006

Stream Started at 1149631208.44
Stream Ended at 1149631211.62
Stream Processed in 3.17 seconds

SQL statements processed: 1
Q14
Begin Execution at Tue Jun 6 17:00:11 2006

-- 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 0.36 seconds.

Ended Executing this Stream at Tue Jun 6 17:00:12 2006

Stream Started at 1149631211.67
Stream Ended at 1149631212.03
Stream Processed in 0.36 seconds

SQL statements processed: 1
Q15
Begin Execution at Tue Jun 6 17:00:12 2006

-- 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 0.49 seconds.

Ended Executing this Stream at Tue Jun 6 17:00:12 2006

Stream Started at 1149631212.08
Stream Ended at 1149631212.57
Stream Processed in 0.49 seconds

SQL statements processed: 1

Appendix E: Seed and Input Parameters

----- qp1.0 -----						
14	1998-01-01					
2	22	STEEL	AFRICA			
9	sienna					
20	metallic	1995-01-01	MOROCCO			
6	1993-01-01	0.04	24			
17	Brand#42	LG JAR				
18	313					
8	FRANCE	EUROPE	PROMO ANODIZED BRASS			
21	IRAQ					
13	pending	deposits				
3	HOUSEHOLD		1995-03-02			
22	10	15	17	14	11	20
	24					
16	Brand#24	MEDIUM ANODIZED	42	7	9	
	32	21	17	6	37	
4	1993-04-01					
11	ALGERIA	0.0000000333				
15	1995-12-01					
1	91					
10	1994-06-01					
19	Brand#23	Brand#34	Brand#33	8	11	20
5	AFRICA	1993-01-01				
7	INDONESIA	FRANCE				
12	AIR	TRUCK	1997-01-01			
----- qp1.1 -----						
21	CANADA					
3	AUTOMOBILE		1995-03-18			
18	314					
5	AMERICA	1993-01-01				
11	JORDAN	0.0000000333				
7	ARGENTINA	RUSSIA				
6	1993-01-01	0.02	25			
20	wheat	1993-01-01	ETHIOPIA			
17	Brand#44	LG PKG				
12	REG AIR	TRUCK	1993-01-01			
16	Brand#54	ECONOMY BURNISHED	41	49	19	
	11	17	40	13	25	
15	1993-09-01					
13	unusual	packages				
10	1993-03-01					
2	9	BRASS	ASIA			
8	RUSSIA	EUROPE	ECONOMY POLISHED BRASS			
14	1993-04-01					
19	Brand#25	Brand#12	Brand#22	3	12	27
9	rosy					
22	17	32	11	23	14	27
	28					
1	99					
4	1995-11-01					
----- qp1.2 -----						
6	1994-01-01	0.07	25			
17	Brand#41	MED CASE				
14	1993-08-01					
16	Brand#34	STANDARD POLISHED	21	20	11	
	4	34	9	40	16	
19	Brand#32	Brand#55	Brand#21	8	13	23
10	1993-12-01					
9	plum					
2	47	NICKEL	AFRICA			
15	1996-04-01					
8	KENYA	AFRICA	ECONOMY BURNISHED BRASS			
5	ASIA	1994-01-01				
22	30	19	14	29	23	18
	33					
12	SHIP	TRUCK	1993-01-01			
7	CHINA	KENYA				
13	unusual	packages				
18	312					
1	107					
4	1993-07-01					
20	honeydew	1997-01-01	SAUDI ARABIA			
3	FURNITURE	1995-03-04				
11	ARGENTINA	0.0000000333				
21	VIETNAM					
----- qp1.3 -----						
8	FRANCE	EUROPE	LARGE BRUSHED BRASS			
5	EUROPE	1994-01-01				
4	1996-02-01					
6	1994-01-01	0.05	24			
17	Brand#43	MED JAR				
7	IRAN	FRANCE				
1	115					
18	313					
22	22	21	19	11	10	12
	26					

14	1993-11-01					
9	orchid					
10	1994-09-01					
15	1993-12-01					
11	KENYA	0.0000000333				
20	saddle	1995-01-01	INDONESIA			
2	35	COPPER	EUROPE			
21	JORDAN					
19	Brand#35	Brand#32	Brand#11	4	14	20
13	unusual	packages				
16	Brand#24	MEDIUM BRUSHED	19	41	15	
	20	30	29	43	4	
12	MAIL	AIR	1993-01-01			
3	MACHINERY		1995-03-20			
----- qp1.4 -----						
5	MIDDLE EAST		1994-01-01			
21	ETHIOPIA					
14	1994-02-01					
19	Brand#32	Brand#25	Brand#15	9	15	27
15	1996-07-01					
17	Brand#45	MED PKG				
12	TRUCK	MAIL	1994-01-01			
6	1994-01-01	0.02	25			
4	1993-11-01					
9	misty					
8	UNITED KINGDOM	EUROPE	LARGE PLATED BRASS			
16	Brand#54	PROMO BURNISHED	22	12	21	
	16	15	44	42	25	
11	BRAZIL	0.0000000333				
2	23	STEEL	AFRICA			
10	1993-06-01					
18	315					
1	62					
13	unusual	packages				
7	BRAZIL	UNITED KINGDOM				
22	17	27	21	30	22	20
	16					
3	FURNITURE	1995-03-06				
20	cyan	1994-01-01	UNITED STATES			
----- qp1.5 -----						
21	RUSSIA					
15	1994-04-01					
4	1996-06-01					
6	1994-01-01	0.08	25			
7	ROMANIA	MOROCCO				
16	Brand#34	SMALL PLATED	31	32	5	
	29	9	36	1	15	
19	Brand#44	Brand#53	Brand#14	4	16	23
18	313					
14	1994-05-01					
22	24	11	20	15	31	17
	28					
11	MOROCCO	0.0000000333				
13	unusual	requests				
3	MACHINERY		1995-03-22			
1	70					
2	10	BRASS	EUROPE			
5	AMERICA	1994-01-01				
8	MOROCCO	AFRICA	LARGE ANODIZED STEEL			
20	orange	1997-01-01	KENYA			
12	RAIL	MAIL	1994-01-01			
17	Brand#42	JUMBO CASE				
10	1994-04-01					
9	magenta					
----- qp1.6 -----						
10	1995-01-01					
3	BUILDING	1995-03-08				
15	1996-11-01					
13	unusual	requests				
6	1995-01-01	0.05	24			
8	GERMANY	EUROPE	MEDIUM POLISHED STEEL			
9	lavender					
7	IRAN	GERMANY				
4	1994-03-01					
11	CANADA	0.0000000333				
22	14	16	19	24	15	11
	34					
18	314					
12	AIR	MAIL	1994-01-01			
1	78					
5	ASIA	1995-01-01				
16	Brand#24	LARGE BRUSHED	35	37	30	
	10	50	13	7	18	
2	48	NICKEL	AMERICA			
14	1994-09-01					
19	Brand#41	Brand#31	Brand#53	9	17	30

```

20      bisque      1995-01-01  EGYPT
17      Brand#44    JUMBO JAR
21      KENYA
----- qp1.7 -----
18      312
8       UNITED STATES      AMERICA      MEDIUM BURNISHED STEEL
20      lemon      1994-01-01  CHINA
21      FRANCE
2       36          TIN          EUROPE
4       1996-10-01
22      23          15          20          27          29          21
22      22
17      Brand#41    JUMBO PKG
1       86
11      MOZAMBIQUE      0.0000000333
9       honeydew
19      Brand#43    Brand#24    Brand#53    5          18          26
3       MACHINERY      1995-03-24
13      unusual      requests
5       EUROPE      1995-01-01
7       BRAZIL      UNITED STATES
10      1993-10-01
16      Brand#54    STANDARD ANODIZED  2          24          19
1       1            46          21          43          7
6       1995-01-01  0.02        25
14      1994-12-01
15      1994-07-01
12      REG AIR      FOB          1994-01-01
----- qp1.8 -----

```

```

19      Brand#55    Brand#52    Brand#52    10          19          23
1       94
15      1997-02-01
17      Brand#43    WRAP CASE
5       MIDDLE EAST      1995-01-01
8       MOZAMBIQUE      AFRICA      SMALL BRUSHED STEEL
9       frosted
12      FOB          RAIL          1997-01-01
14      1995-03-01
7       ROMANIA      MOZAMBIQUE
4       1994-07-01
3       BUILDING      1995-03-10
20      snow          1997-01-01  INDIA
16      Brand#34    MEDIUM PLATED      22          45          7
17      12          13          48          34
6       1995-01-01  0.08        25
22      28          31          16          23          11          12
10      10
10      1994-07-01
13      express      requests
2       24          STEEL      AMERICA
21      UNITED STATES
18      313
11      EGYPT          0.0000000333
----- seed -----
0530221737

```

Appendix F: Benchmark Scripts

```

----- runTPCH -----
#!/bin/ksh
. $KIT_DIR/env

ECHO=echo

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

RUN_ID_FILE=${KIT_DIR}/audit/r_1d

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

RUN_ID=`cat $RUN_ID_FILE`
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

#2starts >> $SCRIPT_LOG_FILE
ckpnt.sh
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}
ckpnt.sh
ckpnt.sh
runTPCHpt ${SCALE_FACTOR} 2 ${RUN_ID}

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

echo "End TPC-H Benchmark SEQUENCE NUMBER: $RUN_ID `date`" >>
$SCRIPT_LOG_FILE
----- runTPCHpt -----
#!/bin/ksh
. $KIT_DIR/env
#ECHO=/bin/echo
SCRIPT_DIR=${KIT_DIR}/scripts
SQL_DIR=${KIT_DIR}/sql
UPD_DIR=${KIT_DIR}/update
SRC_DIR=${KIT_DIR}/utils
QRY_DIR=${KIT_DIR}/queries # this is the location of the query template file
QGEN_DIR=${KIT_DIR}/dbgen
QGEN=${QGEN_DIR}/qgen
QEXEC=${SRC_DIR}

DSS_QUERY=${KIT_DIR}/queries
export DSS_QUERY

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

TPCD_BIN=${KIT_DIR}/audit/bin

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

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

# The defaults

```

```

QPROG=${QEXEC}/qexec

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

set -- `getopt "p:u1:u2:osu:h" "$@"` || usage

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

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

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

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

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

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

    TPCD_LOG_FILE=${TPCD_LOG}/m${PARA}s0
    TPCD_RPT_FILE=${TPCD_RPT}/m${PARA}s0inter
    QRY_FILE=${TPCD_RPT}/qtemp.s${PARA}s0
    QUERY_PARAMETER=${TPCD_LOG}/qp${PARA}.0
    SCRIPT_LOG_FILE=${TPCD_LOG}/m${PARA}stiming
    UF1_LOG=${TPCD_LOG}/m${PARA}s0r1
    UF2_LOG=${TPCD_LOG}/m${PARA}s0r2
    STREAM_COUNT_LOG=${TPCD_LOG}/m${PARA}stsrct

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

    ${QGEN} -c -r `cat $SEED_FILE` -p 0 -s $(SF) -l $QUERY_PARAMETER > ${QRY_FILE}
    START=$GTIME

```

```

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

# Execute UF1

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

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

UF1_END=`$GTIME`
E1DATE=`date`

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

echo "End UF1 $UF1_END, $E1DATE]" >> $$SCRIPT_LOG_FILE
echo UF1: Execution Time: $UF1_TIME >> $$SCRIPT_LOG_FILE
echo >> $$SCRIPT_LOG_FILE

echo "Start Query Part `$GTIME`, `date`" >> $$SCRIPT_LOG_FILE
${QPROG} ${DATABASE_USER} q${QRY_FILE} IS{TPCD_LOG_FILE}
r${TPCD_RPT_FILE} > $DF 2>&1
# Execute UF2

UF2_START=`$GTIME`
E2DATE=`date`

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

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

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

echo "End UF2 $UF2_END, $EDATE]" >> $$SCRIPT_LOG_FILE
echo UF2: Execution Time: $UF2_TIME >> $$SCRIPT_LOG_FILE
echo >> $$SCRIPT_LOG_FILE
echo "End TPC-H Power Test - RUN:${PARA} SEQUENCE:${RUN_ID}, $END, $EDATE"
>> $$SCRIPT_LOG_FILE
MEA_INT=`echo $END - $START | bc`
echo "Elapsed Time for TPC-H Power Test - RUN:${PARA} SEQUENCE:${RUN_ID} is
$MEA_INT" >> $$SCRIPT_LOG_FILE
echo >> $$SCRIPT_LOG_FILE

${KIT_DIR}/audit/abridge.pl ${TPCD_LOG_FILE}
i=$START_SET
PSEED=`cat $SEED_FILE`
while [ $i -le $STOP_SET ]; do
  TPCD_LOG_FILE=${TPCD_LOG}/m${RUN_ID}_$i.log
  TPCD_RPT_FILE=${TPCD_RPT}/m${RUN_ID}_$i.rpt
  QUERY_PARAMETER=${TPCD_LOG}/qp${PARA}.$i
  QRY_FILE=${TPCD_RPT}/qtemp.${PARA}s${i}

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

  i=`expr $i + 1`
done

TH_START_D=`date`
TH_START_T=`$GTIME`
echo >> $$SCRIPT_LOG_FILE
rm -f /tmp/th_pipe1
mknod /tmp/th_pipe1 p
rm -f /tmp/th_pipe2
mknod /tmp/th_pipe2 p
i=$START_SET

echo "Start Throughput Test - RUN:${PARA} SEQUENCE:${RUN_ID} $TH_START_T,
$TH_START_D" >> $$SCRIPT_LOG_FILE
# starts a script to count the streams during the throughput run
(scnt.sh $PARA $RUN_ID > $STREAM_COUNT_LOG &)

while [ $i -le $STOP_SET ]; do
  M_SDATE=`date`
  M_STIME=`$GTIME`
  TPCD_LOG_FILE=${TPCD_LOG}/m${PARA}s${i}

```

```

TPCD_RPT_FILE=${TPCD_RPT}/m${PARA}s${i}inter
echo "Start Query Stream $i $M_STIME, $M_SDATE]" >> $$SCRIPT_LOG_FILE
QRY_FILE=${TPCD_RPT}/qtemp.${PARA}s${i}
${QPROG} ${DATABASE_USER} q${QRY_FILE} IS{TPCD_LOG_FILE}
r${TPCD_RPT_FILE} | grep -v "Connected to ORACLE" >> $$SCRIPT_LOG_FILE &
i=`expr $i + 1`
done

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

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

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

i=$START_SET
while [ $i -le $STOP_SET ]; do
  TPCD_LOG_FILE=${TPCD_LOG}/m${PARA}s${i}
  ${KIT_DIR}/audit/abridge.pl ${TPCD_LOG_FILE}
  i=`expr $i + 1`
done
PIDS=`ps -fu oracle | grep scnt.sh | grep -v grep | awk '{print $2}'`
kill -9 $PIDS
#calculate the metric
#analyze_streams.pl -f p -n $RUN_ID > ${TPCD_RPT}/tpch_metric.${RUN_ID}.$(HID).rpt
-----
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
#
# . SKIT_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=64

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`

sqlplus /NOLOG << !

connect $PASSWD;

```

```

set timing on
set serveroutput on
set echo on

alter session force parallel dml parallel (degree ${PAR_HINT});
alter session set isolation_level = serializable;
alter session set optimizer_index_cost_adj = 10;

insert into orders(
select
o_orderdate      ,
o_orderkey       ,
o_custkey        ,
o_orderpriority  ,
o_shippriority   ,
o_clerk          ,
o_orderstatus    ,
o_totalprice     ,
o_comment
from temp_o_et${SETNUM});

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_linenumbr     ,
l_shipinstruct  ,
l_comment
from temp_l_et${SETNUM});

commit;
exit;
!

END= `SGTIME`

# Done

echo ""
echo "Update Function 1 Set $SETNUM done!"
echo "Elapsed Time is `echo SEND - $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=128
PAR_HINT1=4
SF=${SCALE_FACTOR}
PASSWD=${DATABASE_USER}

if [ $# -lt 1 ]
then

```

```

usage
exit 1
fi

SETNUM=${1}

i=1
PID=""

START=`SGTIME`
# first create the temp tables

sqlplus /NOLOG << !

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

create table temp_okey${SETNUM} tablespace TSD1 parallel ${PAR_HINT1} nologging as
select * from temp_okey_et${SETNUM});

create unique index i_temp_okey${SETNUM} on temp_okey${SETNUM} (t_orderkey)
tablespace TSD1 parallel ${PAR_HINT1} nologging compute statistics;

analyze table temp_okey${SETNUM} estimate statistics sample 1 percent;

alter session force parallel dml parallel ${PAR_HINT};
alter session set isolation_level=serializable;
alter session set optimizer_index_cost_adj = 10;

delete from (select /*+ use_nl(o) */ o.rowid from orders o, temp_okey${SETNUM} t where
o.o_orderkey = t.t_orderkey order by 1);

delete from (select /*+ use_nl(l) */ l.rowid from lineitem l,temp_okey${SETNUM} t where
l.l_orderkey = t.t_orderkey order by 1);

commit;
exit;
!

END= `SGTIME`

# Done

echo ""
echo "Update Function 2 Set $SETNUM done!"
echo "Elapsed Time is `echo SEND - $START | bc`"
----- qexecplc -----
#ifdef RCSID
static char *RCSid =
    "$Header: qexecplc 17-oct-2001.09:29:47 mpoess Exp $ ";
#endif /* RCSID */

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

/*
NAME
qexecplc - <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 gettimeofday();

/* 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 SQLlimit();
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 */

slist[MAX_SEL_LIST]; /* Array for describing Select List */
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 cmmnt[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 pfinem = PFMEMSIZE; /* Memory to prefetch rows */

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

/* OCI handles */
OCIEnv *tpcenv = NULL;
OCIServer *tpcsv = NULL;
OCIError *errhp = NULL;
OCISvcCtx *tpscvc = NULL;
OCISession *tpcusr = NULL;
OCISmt *curq = NULL;
OCISmt *cur_dml = NULL;
OCISmt *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");
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;

```

```

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(tpscvc,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", ctime(&tim));
fprintf(rep, "Begin Executing this Stream at %s\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",
              num_to_fetch);
      break;

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

```

```

/* Executes the query */
SQLExec();

stmt_cnt++;
qry_cnt++;
flush(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);
flush(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", (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", (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",
        (tr_end - tr_start));

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

flush(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] = (dlttype *) memalloc (sizeof(dlttype));
      dlist[i]->defhdl = NULL;
      /* OCThalloc(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);

OCIAlloc(tpcenv,&errhp,OCI_HTYPE_ERROR);
OCIAlloc(tpcenv,&curq,OCI_HTYPE_STMT);
OCIAlloc(tpcenv,&cur_dml,OCI_HTYPE_STMT);
OCIAlloc(tpcenv,&cur_ddl,OCI_HTYPE_STMT);
OCIAlloc(tpcenv,&tpcsvc,OCI_HTYPE_SVCCTX);
OCIAlloc(tpcenv,&tpcsrv,OCI_HTYPE_SERVER);
OCIAlloc(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", 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 = gettimeofday();
fprintf(logfile,"Query Processed in %.2f seconds.\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 %.2fn", qnp,(s_tr_end - s_tr_start));*/
fprintf(rep, "Query %s: Execution Time: %.2f started %.2f ended %.2fn",
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 = gettimeofday();

```

```

s_tr_start = gettimeofday();

/* prepare the statement */

if ((status = OCISmtPrepare(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 this version */

OCIaset(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, &pfimem, 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 statements will screw it up */

OCIaset(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);
OCIIFREE(tpcenv,OCI_HTYPE_STMT);
OCIIFREE(tpcenv,OCI_HTYPE_SVCCTX);
OCIIFREE(tpcenv,OCI_HTYPE_SERVER);
OCIIFREE(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] = (dtype *) memalloc(sizeof(dtype));
            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 = OCISmtFetch(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 = OCISmtFetch(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 = OCISmtFetch(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 {
                OCISmtFetch(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 (strcmp(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 = '\0';
}

----- qexecplh -----
/*
* $Header: qexecplh 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
qexecplh

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

/* some basic definitions */

# define UNAME_LEN 64
# define MAX_FILE_PATH_LEN 128

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

# ifndef FALSE
# define FALSE 1
# endif /* FALSE */
# ifndef LINUX
# define MAX(x,y) ((x >= y) ? x : y)
# define MIN(x,y) ((x <= y) ? x : y)
# endif
/* defines and typedefs for parsing */

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

/* defines and typedefs for query description */

# define MAX_COLNAME_SIZE 32 /* Maximum length of Column name */
# define MAX_SEL_LIST 16 /* Maximum items on a select list */

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

/* types for describe */

# define CHAR_TYPE 1
# define NUM_TYPE 2
# define INT_TYPE 3
# define FLT_TYPE 4

```

```

# define STR_TYPE 5
# define DATE_TYPE 12

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

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

typedef struct des
{
    ub2 dbsize;
    ub4 buflen;
    /* sb2 dsize; */
    sb4 scale;
    /* sb2 nullok; */
    OCITypeCode dbtype;
    /* text buf[MAX_COLNAME_SIZE]; */
    text *buf;
    ub1 precision;
} sltype;

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

# define MAX_ARRAY 50 /* Maximum array size for array fetch */
# define PFMEMSIZE 65536 /* Memory size of prefetch buffer */

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

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

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

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

extern int errno;

# define SQL_LEN 2048

# ifndef NULL
# define NULL 0
# endif

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

# ifndef DISCARD
# define DISCARD (void)
# endif

# ifndef sword
# define sword int
# endif

# ifndef ub1
# define ub1 unsigned char
# endif

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

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

/* For get_statement */

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

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

```

```

DISCARD 0

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

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

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

```

```

#define OCIsexec(svch,stmh,errh,iter) \
if((status=OCISmtExecute(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 PDDLTX "alter session force parallel ddl parallel (degree 84)"

#endif /* QSTREAMPL_H */

```

Appendix G: Price Quotes

From: Mary.Beth Pierantoni
[mary.beth.pierantoni@oracle.com]
Sent: Tuesday, June 06, 2006 6:07 PM
To: Georgson, Bryon
Subject: Oracle Pricing

Product	Price	Quantity	Extended Price
Oracle Database 10g Enterprise Edition Release 2, Named User Plus for 3 years	10,000	64*	640,000
Oracle Real Application Clusters	5,000	64*	320,000
Partitioning	2,500	64*	160,000
Database Server Support Package	128,000	3	384,000
Oracle Mandatory E-Business Discount			<376,000>
TOTAL			1,128,000

Oracle Pricing Contact: MaryBeth Pierantoni,
mary.beth.pierantoni@oracle.com, 916-315-5081

** 64 = 128 * 0.50. Explanation: For the purposes of counting the number of processors which require licensing, an AMD multicore chip with "n" cores shall be determined by multiplying "n" cores by a factor of 0.50.