
HP AlphaServer ES45 Model 68/1000
using
HP Tru64 UNIX V5.1A/IPK
and
***Oracle9i Database Enterprise Edition Release 2 with Real
Application Clusters and Partitioning Options***

TPC Benchmark™ H
Full Disclosure Report



i n v e n t

ORACLE®

Second Edition – November 18, 2002

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 Hewlett-Packard Company 2002.

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.

Printed in the U.S.A., November 18, 2002.

AlphaServer, Tru64 UNIX and the HP logo are registered trademarks of the Hewlett-Packard Company.

Oracle9i and the Oracle logo are trademarks of Oracle Corporation.

TPC Benchmark and TPC-H are registered 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.

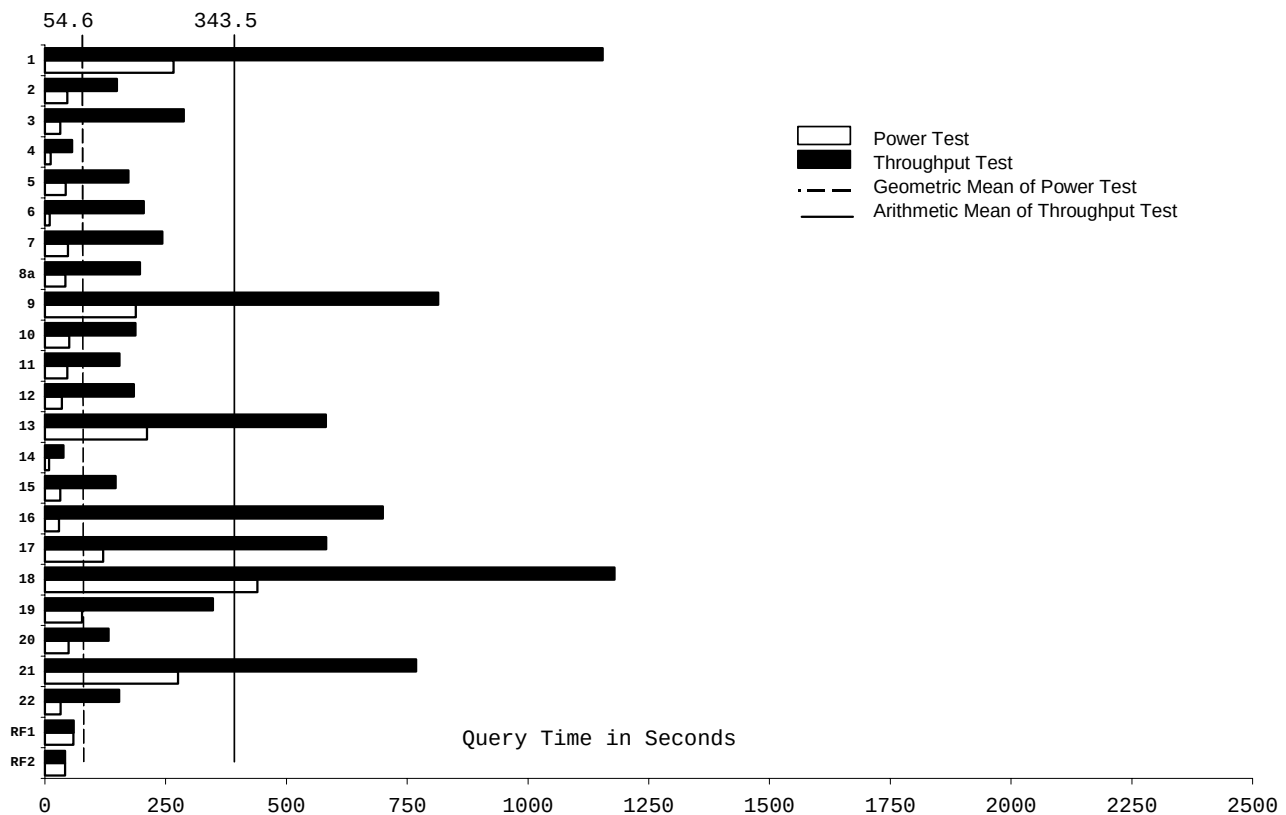


HP AlphaServer ES45 Cluster Using Oracle9i Release 2 with Real Application Clusters

TPC-H Rev. 2.0

Report Date:
November 18, 2002

Total System Cost		Composite Query per Hour Metric		Price / Performance
\$1,991,562		5578.4 QphH@100GB		\$357.01 \$/QphH@100GB
Database Size	Database Manager	Operating System	Other Software	Availability Date
100GB	Oracle9i Enterprise Edition Release 2 with Real Application Clusters	HP Tru64 UNIX V5.1A/IPK	None	July 15, 2002



Database Load Time = 1:10:42	Load Included Backup: N	Total Data Storage / Database Size = 56.48
RAID (Base tables only): N	RAID (Base tables and auxiliary data structures): N	RAID (All): Y

System Configuration: (4 Node Cluster Aggregate)

Processors: 16 Alpha EV 68/1000 MHz CPUs with 8 MB cache

Memory: 128 GB memory

Disks: 336 Drives

Total Disk Storage: 5,647.7 GB



Compaq AlphaServer ES45 Cluster Using Oracle9i Release 2 with Real Application Clusters

TPC-H Rev. 2.0

Report Date:
November 18, 2002

Description	Part Number	Third Party Brand	Unit Price	Qty	Extended Price	3 yr. Maint. Price
Server Hardware						
Compaq AS ES45 68/1000 M2 4GB Unix	DA-68DAA-DA	1	50,809	4	203,236	71,584
ES45 Tower Enclosure	BA61M-CT	1	325	4	1,300	Inc.
ES45 68/1000 SMP CPU Unix	KN610-DB	1	10,400	12	124,800	Inc.
ES45 4GB Memory Option	MS620-DA	1	17,306	28	484,568	Inc.
Optional Power Supply, Self Sensing	H7906-A9	1	813	4	3,252	Inc.
18.2GB 10K RPM Ultra3 SCSI	3R-A0585-AA	1	207	13	2,691	Inc.
Power Cord	BN18J-1K	1	8	8	64	Inc.
SCSI Drive Cage, 6 Slot	BA610-6D	1	488	4	1,952	Inc.
1 CH Wide Ultra-2 (LVD) Adapter	3X-KZPCA-AA	1	246	4	984	Inc.
PCI to Dual-Prt FE TX NIC	3X-DE602-AA	1	246	4	984	Inc.
Memory Channel Hub (w/4 line cards)	CCMHB-AA	1	6,365	1	6,365	Inc.
PCI MC2 Controller	CCMAB-AA	1	1,944	4	7,776	Inc.
10 Meter Cable for MC2	BN39B-10	1	273	4	1,092	Inc.
64 Bit FC HBA T64/VMS	DS-KGP5A-CA	1	2,209	16	35,344	Inc.
D Shelf 180W 1Doc BLW Metric Blue	DS-BA356-JD	1	919	2	1,838	Inc.
PCI to UltraSCSI Adapter UW SE	KZPBA-CA	1	299	2	598	Inc.
Ultra 68VHD 3M Cable Assembly	BN37A-03	1	91	2	182	Inc.
VT510;W hite; North Amer; No key	VT510-AA	1	397	4	1,588	0
US/CANADA W95 KYBD WHIT	PCXLA-NA	1	16	4	64	0
Subtotal					878,678	71,584
STORAGE:						
USA Model 914 Storage Shelf	DS-SW XEB-AA	1	46,406	4	185,624	68,424
Controller w/o ECB Cable Kit	DS-HSG80-BK	1	8,793	24	211,032	Inc.
16-Port SAN Switch	DS-DSGGB-AB	1	18,525	4	74,100	Inc.
Short-W ave Optical GBICs	DS-DXGGA-SA	1	241	40	9,640	Inc.
Model 2200 ECB	DS-SE2CS-CB	1	369	24	8,856	Inc.
Fibre Channel Cables	BNGBX-30	1	246	40	9,840	Inc.
12/24gb 4MM Dat 5.25 Tape Drive	TLZ10-LB	1	444	4	1,776	3,448
9.1GB 7200RPM Disk**	DS-RZ1DA-VW	1	423	6	2,538	Spared
18.2GB 10K RPM Ultra3 SCSI	3R-A0585-AA	1	206	317	65,302	Spared
Subtotal					568,708	71,872
Software						
3YR, AS ES40/45 UNIX BRNZ24X7	FM-E4WUS-36	1	1,896	4		7,584
3YR Digital Unix O/S & LP	FM-CDDST-36	1	6,765	4		27,060
TRU64 UNIX AlphaCDROM	QA-MT4AA-H8	1	293	4	1,172	
3YR AS ES45 UNIX SMP	FM-62USM-36	1	214	12		2,568
3YR 7X24 HS*80 Platform SW	FM-PLAT2-36	1	414	24		9,936
HSG80 ACSsf All Lic/PCRM Pkg	QB-6BUAA-SB	1	4,550	24	109,200	
Oracle9i Database Enterprise Edition Release 2, Named User Plus for 3 years	Oracle	2	10,000	16	160,000	
Real Application Clusters, Named User Plus for 3 Years	Oracle	2	5,000	16	80,000	
Partitioning, Named User Plus for 3 years	Oracle	2	2,500	16	40,000	
Oracle DB Server Support Pkg. for 3 years	Oracle	2	6,000	4		24,000
Oracle Mandatory E-Business Discount (license and support)*	Oracle	2	(\$60,800)	1	(\$60,800)	
Subtotal					329,572	71,148

Notes:** 10% Spares

1=IC System Solutions 2=Oracle(Herve
Lejeune; herve.lejeune@oracle.com 650 560-
1894 (Please see Appendix G))

Subtotal	\$1,776,958	\$214,604
Three-Year Cost of Ownership:	\$1,991,562	
QphH	5,578.40	
\$ / QphH	\$357.01	

Audited by InfoSizing

All prices are based on similar quantities and configurations.

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



Compaq AlphaServer ES45 Cluster Using Oracle9i Release 2 with Real Application Clusters

TPC-H Rev. 2.0

Report Date:
November 18, 2002

Numerical Quantities

Measurement Results:

Database Scale Factor	= 100
Total Data Storage / Database Size	= 56.48
Start of Database Load	= 2002-06-24 16:53:24
End of Database Load	= 2002-06-24 18:04:06
Database Load Time	= 01:10:42
Query Streams for Throughput Test	= 5
TPC-H Power	= 6596.2 @100GB
TPC-H Throughput	= 4717.7 @100GB
TPC-H Composite Query-per-Hour Metric (<u>QphH@Size</u>)	= 5578.4 @100GB
Total System Price Over 5 Years	= \$1,991,562
TPC-H Price/ Performance Metric (<u>\$/QphH@Size</u>)	= \$357.01

Measurement Intervals:

Measurement Interval in Throughput Test (Ts)	= 8,394.0 seconds
--	-------------------

Duration of Stream Execution:

Stream ID	Seed	Start Date	Start Time	End Date	End Time	Duration
Stream 0:	624180406	6/24/2002	21:00:31	6/24/2002	21:37:08	0:36:37
Stream 1:	624180407	6/24/2002	21:37:10	6/24/2002	23:41:28	2:04:18
Stream 2:	624180408	6/24/2002	21:37:10	6/24/2002	23:48:35	2:11:25
Stream 3:	624180409	6/24/2002	21:37:10	6/24/2002	23:44:08	2:06:58
Stream 4:	624180410	6/24/2002	21:37:10	6/24/2002	23:43:31	2:06:21
Stream 5:	624180411	6/24/2002	21:37:10	6/24/2002	23:37:56	2:00:46
Refresh		6/24/2002	23:48:35	6/24/2002	23:57:04	0:08:29



HP AlphaServer ES45 Cluster Using Oracle9i Release 2 with Real Application Clusters

TPC-H Rev. 2.0

Report Date:
November 18, 2002

TPC-H Timing Intervals (in seconds)

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8a	Q9	Q10	Q11	Q12
Stream 00	266.4	46.3	31.5	11.9	43.2	9.9	47.7	42.1	188.3	50.4	46.6	35.4
Stream 01	1144.2	151.9	122.3	36.7	217.3	46.9	231.9	179.7	757.9	198.4	193.7	119.1
Stream 02	1452.4	138.2	83.6	55.9	134.0	36.4	284.7	230.2	1092.2	154.1	91.4	240.4
Stream 03	941.1	144.2	77.7	48.1	167.4	39.1	243.1	189.7	827.7	250.6	188.5	118.6
Stream 04	1133.1	153.5	80.9	69.8	141.1	34.2	234.6	220.2	745.2	189.4	132.1	201.2
Stream 05	1102.9	157.7	157.9	71.0	205.7	53.6	220.7	164.7	646.9	146.1	165.6	241.8
Minimum	941.1	138.2	77.7	36.7	134.0	34.2	220.7	164.7	646.9	146.1	91.4	118.6
Average	1154.7	149.1	287.9	56.3	173.1	204.6	243.0	196.9	814.0	187.7	154.3	184.2
Maximum	1452.4	157.7	157.9	71.0	217.3	53.6	284.7	230.2	1092.2	250.6	193.7	241.8

	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	RF1	RF2
Stream 00	211.3	8.9	31.6	29.7	120.8	440.0	76.9	48.6	275.9	32.5	59.0	42.0
Stream 01	687.6	21.6	81.7	229.8	354.9	1212.4	268.7	156.7	916.3	128.6	62.0	44.0
Stream 02	672.5	52.2	181.4	175.1	894.2	1019.7	354.6	126.0	287.7	128.6	57.0	38.0
Stream 03	499.9	45.4	181.7	114.4	783.9	1135.6	385.7	137.2	848.9	249.5	59.0	44.0
Stream 04	640.7	38.2	183.8	123.2	457.4	1372.9	365.9	80.0	859.3	124.1	59.0	42.0
Stream 05	408.9	33.3	102.6	200.7	421.4	1154.6	364.5	158.8	928.2	136.9	61.0	42.0
Minimum	408.9	21.6	81.7	114.4	354.9	1019.7	268.7	80.0	287.7	124.1	57.0	38.0
Average	581.9	38.1	146.2	699.6	582.4	1179.0	347.9	131.7	768.1	153.5	59.6	42.0
Maximum	687.6	52.2	183.8	229.8	894.2	1372.9	385.7	158.8	928.2	249.5	62.0	44.0

Table of Contents

TABLE OF CONTENTS	VII
1 GENERAL ITEMS	11
1.1 BENCHMARK SPONSOR.....	11
1.2 PARAMETER SETTINGS	11
1.3 CONFIGURATION ITEMS	11
2 CLAUSE 1: LOGICAL DATABASE DESIGN.....	13
2.1 TABLE DEFINITIONS.....	13
2.2 PHYSICAL ORGANIZATION OF DATABASE	13
2.3 HORIZONTAL PARTITIONING.....	13
2.4 REPLICATION	13
3 CLAUSE 2: QUERIES AND UPDATE FUNCTIONS.....	14
3.1 QUERY LANGUAGE.....	14
3.2 RANDOM NUMBER GENERATION	14
3.3 SUBSTITUTION PARAMETERS GENERATION	14
3.4 QUERY TEXT AND OUTPUT DATA FROM DATABASE.....	14
3.5 QUERY SUBSTITUTION PARAMETERS AND SEEDS USED.....	14
3.6 QUERY ISOLATION LEVEL	14
3.7 REFRESH FUNCTION IMPLEMENTATION	15
4 CLAUSE 3: DATABASE SYSTEM PROPERTIES.....	16
4.1 ACID PROPERTIES	16
4.2 ATOMICITY	16
4.2.1 <i>Completed Transaction</i>	16
4.2.2 <i>Aborted Transaction</i>	16
4.3 CONSISTENCY.....	17
4.3.1 <i>Consistency Test</i>	17
4.4 ISOLATION.....	17
4.4.1 <i>Read-Write Conflict with Commit</i>	17
4.4.2 <i>Read-Write Conflict with Rollback</i>	17
4.4.3 <i>Write-Write Conflict with Commit</i>	18
4.4.4 <i>Write-Write Conflict with Rollback</i>	18
4.4.5 <i>Concurrent Read/Write Transactions on Different Tables</i>	18
4.4.6 <i>Read-Only Query Conflict with Update Transaction</i>	19
4.5 DURABILITY	19
4.5.1 <i>Failure of a Durable Medium</i>	19
4.5.2 <i>System Crash</i>	19
4.5.3 <i>Memory Failure</i>	19
5 CLAUSE 4: SCALING AND DATABASE POPULATION	20
5.1 CARDINALITY OF TABLES.....	20
5.2 DISTRIBUTION OF TABLES AND LOGS ACROSS MEDIA.....	20
5.3 PARTITIONING AND REPLICATION	21
5.4 RAID FEATURE	21
5.5 RAID INTENT.....	21
5.6 DBGEN VERSION AND MODIFICATIONS	21
5.7 DATABASE LOAD TIME.....	21
5.8 DATA STORAGE RATIO	22
5.9 DATABASE LOAD MECHANISM DETAILS AND ILLUSTRATION	22
5.10 QUALIFICATION DATABASE CONFIGURATION	22

6	CLAUSE 5: PERFORMANCE METRICS AND EXECUTION RULES	23
6.1	SYSTEM ACTIVITY BETWEEN LOAD AND PERFORMANCE TESTS	23
6.2	STEPS IN THE POWER TEST	23
6.3	TIMING INTERVALS.....	23
6.4	NUMBER OF STREAMS FOR THE THROUGHPUT TEST	23
6.5	START/FINISH TIME OF EACH QUERY STREAM	23
6.6	TOTAL ELAPSED TIME.....	24
6.7	START/FINISH TIME FOR THE REFRESH FUNCTION	24
6.8	TIMING INTERVALS FOR EACH QUERY AND REFRESH FUNCTION FOR EACH STREAM.....	24
6.9	PERFORMANCE METRICS	24
6.10	PERFORMANCE METRIC AND NUMERICAL QUANTITIES FROM BOTH RUNS.....	24
6.11	SYSTEM ACTIVITY BETWEEN PERFORMANCE RUNS.....	25
7	CLAUSE 6: SUT AND DRIVER IMPLEMENTATION	26
7.1	DRIVER.....	26
7.2	IMPLEMENTATION-SPECIFIC LAYER.....	26
7.3	PROFILE-DIRECTED OPTIMIZATION.....	26
8	CLAUSE 7: PRICING.....	27
8.1	HARDWARE AND SOFTWARE USED	27
8.2	TOTAL THREE YEAR PRICING	27
8.3	AVAILABILITY DATE.....	27
9	CLAUSE 9: AUDIT RELATED ITEMS.....	29
9.1	AUDITOR'S REPORT & ATTESTATION LETTER.....	29
	APPENDIX A: TRU64 UNIX AND ORACLE9I PARAMETERS	31
	ORACLE9I PARAMETERS	31
	<i>common.ora</i>	31
	<i>dbcreate.ora</i>	31
	<i>tpchorc1.ora (Node 1)</i>	31
	<i>tpchorc2.ora (Node 2)</i>	31
	<i>tpchorc3.ora (Node 3)</i>	31
	<i>tpchorc4.ora (Node 4)</i>	31
	ORACLE9I ENVIRONMENT VARIABLES	31
	<i>tpch.env.oracle</i>	31
	TRU64 UNIX PARAMETERS	32
	<i>sysconfig_tpchorc1 (Node 1)</i>	32
	<i>sysconfig_tpchorc2 (Node 2)</i>	33
	<i>sysconfig_tpchorc3 (Node 3)</i>	34
	<i>sysconfig_tpchorc4 (Node 4)</i>	35
	APPENDIX B: SCRIPTS AND PROGRAMS	37
	DATABASE CREATE AND LOAD	37
	<i>create_database_4node.sh</i>	37
	<i>create_tablespace_lineitem.sql</i>	37
	<i>create_tablespace_orders.sql</i>	38
	<i>create_tablespace_schema.sql</i>	38
	<i>create_tables_all_et.sql</i>	39
	<i>dbgen_flatfiles.sh</i>	40
	<i>shutdown_instance.sh</i>	41
	<i>startup_4_instances.sh</i>	41
	<i>startup_tpchorc1.sh</i>	41
	<i>startup_tpchorc2.sh</i>	42
	<i>startup_tpchorc3.sh</i>	42
	<i>startup_tpchorc4.sh</i>	42

<i>database_load_tables_et.sh</i>	42
<i>create_indexes_all.sh</i>	45
<i>analyze_ops.sh</i>	45
ACID TEST SOURCE CODE	45
<i>atom.sh</i>	45
<i>atranspl.c</i>	47
<i>atranspl.h</i>	51
<i>ckpt.sh</i>	52
<i>cnt_hist.sql</i>	52
<i>consist.sh</i>	52
<i>consist.sql</i>	53
<i>count_tx.sh</i>	54
<i>d_hist.sql</i>	54
<i>end_acid.sh</i>	54
<i>gettime.c</i>	55
<i>iso1.sh</i>	58
<i>iso2.sh</i>	58
<i>iso3.sh</i>	59
<i>iso4.sh</i>	60
<i>iso5.sh</i>	61
<i>iso6.sh</i>	61
<i>randkey.c</i>	62
<i>randpsup.c</i>	64
<i>run_acid.sh</i>	65
<i>sample.sh</i>	65
<i>sample.sql</i>	66
DISK CONFIGURATION DATA	66
<i>HSG80_storage_array_configuration</i>	66
LOGICAL STORAGE MANAGER SCRIPTS	66
<i>lsm_create.sh</i>	66
<i>volsetup_disks.sh</i>	66
<i>sd_files.sh</i>	66
<i>sd_flatfiles.sh</i>	67
<i>sd_lineitem.sh</i>	67
<i>sd_orders.sh</i>	67
<i>sd_temps.sh</i>	67
<i>plex_files.sh</i>	67
<i>plex_flatfiles.sh</i>	68
<i>plex_lineitem.sh</i>	68
<i>plex_orders.sh</i>	68
<i>plex_temps.sh</i>	68
<i>prepare4acid.sh</i>	69
<i>create_100gb_tpch_links.sh</i>	69
APPENDIX C: QUERY TEXT AND RESULT OUTPUT	71
<i>mqs00q01</i>	71
<i>mqs00q02</i>	71
<i>mqs00q03</i>	74
<i>mqs00q04</i>	75
<i>mqs00q05</i>	75
<i>mqs00q06</i>	75
<i>mqs00q07</i>	75
<i>mqs00q08</i>	76
<i>mqs00q09</i>	76
<i>mqs00q10</i>	78
<i>mqs00q11</i>	79

<i>mqs00q12</i>	80
<i>mqs00q13</i>	80
<i>mqs00q14</i>	81
<i>mqs00q15</i>	81
<i>mqs00q16</i>	82
<i>mqs00q17</i>	83
<i>mqs00q18</i>	83
<i>mqs00q19</i>	84
<i>mqs00q20</i>	84
<i>mqs00q21</i>	85
<i>mqs00q22</i>	87
APPENDIX D: SEED VALUES AND QUERY SUBSTITUTION PARAMETERS	89
<i>seed_values</i>	89
<i>m2param.0</i>	89
<i>m2param.1</i>	89
<i>m2param.2</i>	89
<i>m1param.3</i>	89
<i>m2param.4</i>	89
<i>m2param.5</i>	90
APPENDIX E: IMPLEMENTATION-SPECIFIC LAYER/DRIVER CODE	91
<i>runTPCHall</i>	91
<i>runTPCHpt</i>	91
<i>runTPCHus</i>	93
<i>runuf1.sh</i>	94
<i>runuf2.sh</i>	95
<i>shutdown_instance.sh</i>	95
<i>startup_4_instances.sh</i>	95
<i>startup_tpchorc1.sh</i>	96
<i>startup_tpchorc2.sh</i>	96
<i>startup_tpchorc3.sh</i>	96
<i>startup_tpchorc4.sh</i>	96
<i>env</i>	96
<i>gtime.c</i>	96
<i>qexecpl.c</i>	96
<i>qexecpl.h</i>	102
APPENDIX F	105
<i>checkidx.sql</i>	105
<i>dbtables.sql</i>	105
<i>firstten.sql</i>	106
APPENDIX G - PRICING.....	107

1 General Items

1.1 Benchmark Sponsor

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

Hewlett-Packard Company and Oracle Corporation are the sponsors of this TPC-H benchmark.

1.2 Parameter Settings

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

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

Appendix A contains the tunable parameters for the database and operating system.

1.3 Configuration Items

Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences.

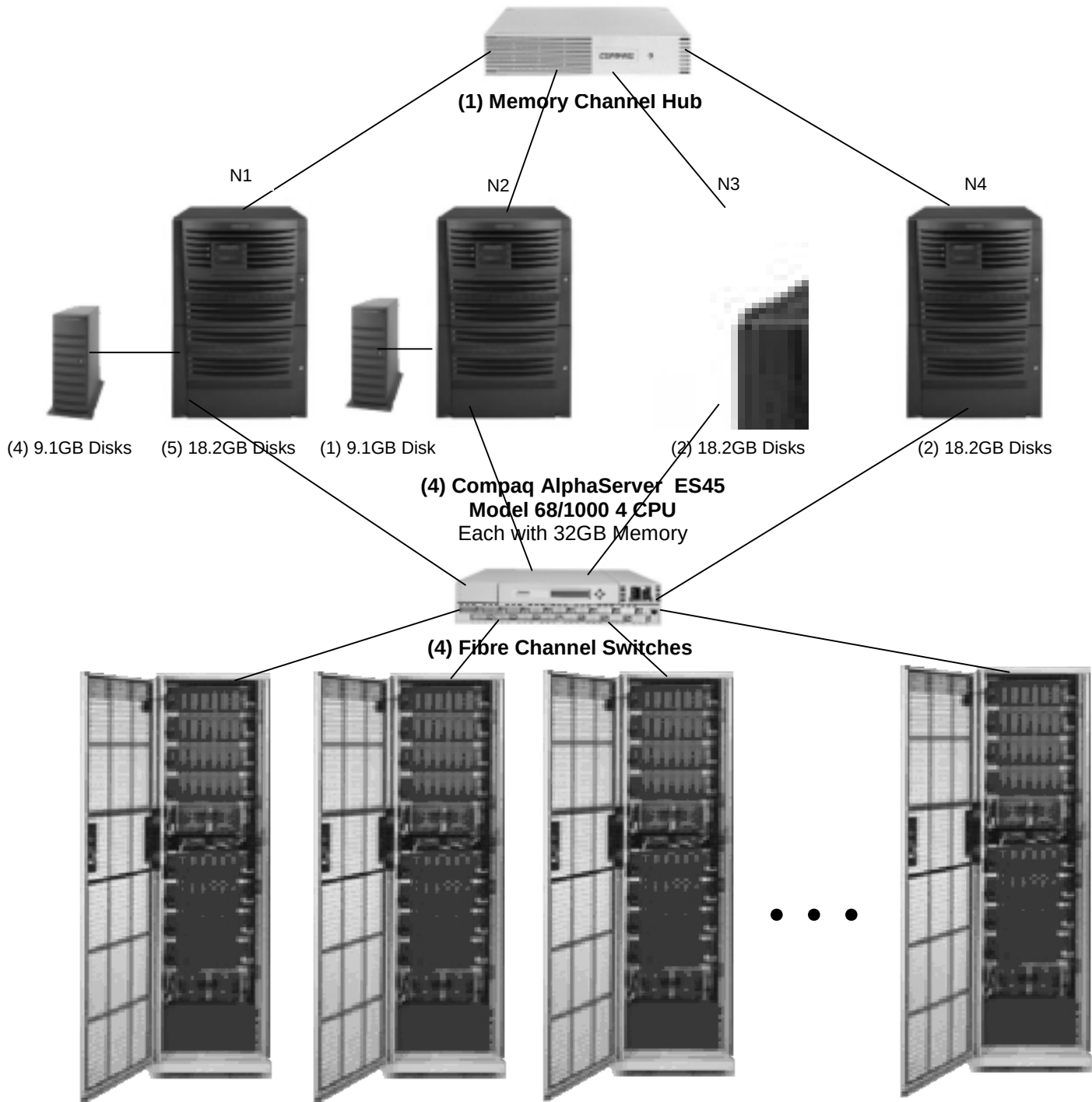
The benchmark system is depicted on the following page and consists of (4) ES45 Model 68/1000 nodes each containing:

- (1) Internal SCSI Controller
- (4) KGPSA-CA PCI SCSI Controllers
- (1) 10/100 Base-T Ethernet Adapter
- (1) Memory Channel Interconnect Adapters
- (1) Qlogic SCSI Controller (Node 1 only)

Cluster wide:

- (1) Memory Channel Hubs with Line Cards
- (4) I/O Fibre Channel Switches
- (24) HSG80 Disk Controllers (4 Full Cabinets)
- (5) DS-RZ1DA-VW (9.1GB 7200 RPM Disk Drives)
- (297) 18.2GB 10K RPM Ultra3 SCSI Disk Drives

System Configuration Diagram (Measured and Priced)



2 Clause 1: Logical Database Design

2.1 Table Definitions

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

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

2.2 Physical Organization of Database

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

No record clustering or index clustering were used. Column ordering was changed for some tables. Refer to the table create statements in Appendix B for further details.

2.3 Horizontal Partitioning

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

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

Replication was not used for this benchmark.

3 Clause 2: Queries and Update Functions

3.1 Query Language

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

SQL was the query language used to implement all queries.

3.2 Random Number Generation

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

The TPC supplied version 1.3.0 of DBGEN and QGEN were used for this benchmark.

3.3 Substitution Parameters Generation

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

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

3.4 Query Text and Output Data from Database

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

1) Appendix C contains the query text and query 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 D contains the seed and query substitution parameters.

3.6 Query Isolation Level

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

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

3.7 Refresh Function Implementation

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 a part of the driver code. See Appendix E.

4 Clause 3: Database System Properties

4.1 ACID Properties

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

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

4.2 Atomicity

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

4.2.1 Completed Transaction

Perform the ACID Transaction (see Clause 3.1.5) for a randomly selected set of input data and verify that the appropriate rows have been changed in the ORDERS, LINEITEM, and HISTORY tables.

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

4.2.2 Aborted Transaction

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

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

4.3 Consistency

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

4.3.1 Consistency Test

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

1. The consistency of the ORDER and LINEITEM tables was verified based on a sample of O_ORDERKEYs.
2. 100 ACID transactions were submitted from each of 6 execution streams.
3. The consistency of the ORDER and LINEITEM tables was verified a second time.

4.4 Isolation

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

4.4.1 Read-Write Conflict with Commit

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

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

4.4.2 Read-Write Conflict with Rollback

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

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

4.4.3 Write-Write Conflict with Commit

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

1. An ACID transaction, T1, was started for a randomly selected O_KEY, L_KEY, and DELTA. T1 was suspended prior to commit.
2. Another ACID transaction, T2, was started using the same O_KEY and L_KEY and a randomly selected DELTA.
3. T2 waited.
4. T1 was allowed to COMMIT and T2 completed.
5. It was verified that $T2.L_EXTENDEDPRICE = T1.L_EXTENDEDPRICE + (DELTA1 * (T1.L_EXTENDEDPRICE / T1.L_QUANTITY))$.

4.4.4 Write-Write Conflict with Rollback

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

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

4.4.5 Concurrent Read/Write Transactions on Different Tables

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

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

For random values of PS_PARTKEY and PS_SUPPKEY, all columns of the PARTSUPP table for which PS_PARTKEY and PS_SUPPKEY are equal, are returned.
3. T2 completed.
4. T1 was allowed COMMIT.

5. It was verified that appropriate rows in ORDERS, LINEITEM and HISTORY tables were changed.

4.4.6 Read-Only Query Conflict with Update Transaction

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

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

4.5 Durability

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

4.5.1 Failure of a Durable Medium

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

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

4.5.2 System Crash

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

- 1) The system crash and memory failure tests were combined. Power to both servers was turned off simultaneously by flipping breakers at the main electrical panel during the durability test. When power was restored, the system rebooted and the database was restarted. The durability success file and the HISTORY table were compared successfully.

4.5.3 Memory Failure

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

See section 4.5.2.

5 Clause 4: Scaling and Database Population

5.1 Cardinality of Tables

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

TABLE	ROWS
ORDERS	150,000,000
LINEITEM	600,037,902
CUSTOMER	15,000,000
PART	20,000,000
SUPPLIER	1,000,000
PARTSUPP	80,000,000
NATION	25
REGION	5

5.2 Distribution of Tables and Logs Across Media

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

- All table, index, log and temp data and mirrors were striped across 12 Compaq HSG80 storage controller arrays, each controller array supporting 12 mirrored pairs of disks.
- The following disks were serviced by the internal SCSI controller:

Node 1:

- One disk for the operating system (standalone boot)
- One disk for the operating system (cluster boot)
- One disk for the operating system (cluster member boot)
- One disk for Oracle
- One disk for /oracle_save
- One spare disk

Nodes 2-4:

- One disk for the operating system (member boot)
- One spare disk

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

5.3 Partitioning and Replication

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

The database was not replicated.

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

5.4 RAID Feature

Implementations may use some form of RAID . The RAID level used must be disclosed for each device.

Hardware RAID 1 + 0 (mirroring and striping) was used to store data on log disks and all other database disks.

5.5 RAID Intent

If RAID is used in an implementation, the logical intent of its use must be disclosed. Three levels of usage are defined:

- 1. Base tables only: In this case only the Base Tables (see Clause 1.2) are protected by any form of RAID;*
- 2. Base tables and auxiliary data structures: in addition to the protection of the base tables, implementations in this class must also employ RAID to protect all auxiliary data structures;*
- 3. Everything: implementations in this usage category must employ RAID to protect all database storage, including temporary or scratch space in addition to the base tables and auxiliary data structures.*

The level of intent for RAID was "Everything".

5.6 DBGEN Version and Modifications

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

The supplied DBGEN version 1.3.0 was used to generate the database population for this benchmark.

5.7 Database Load Time

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

The database load time was 1:10:42.

5.8 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 Clause 4.1.3.1. The ratio must be reported to the nearest 1/100th, rounded up. For example, a system configured with 96 disks of 2.1 GB capacity for a 300GB test database has a data storage ratio of 2.02.

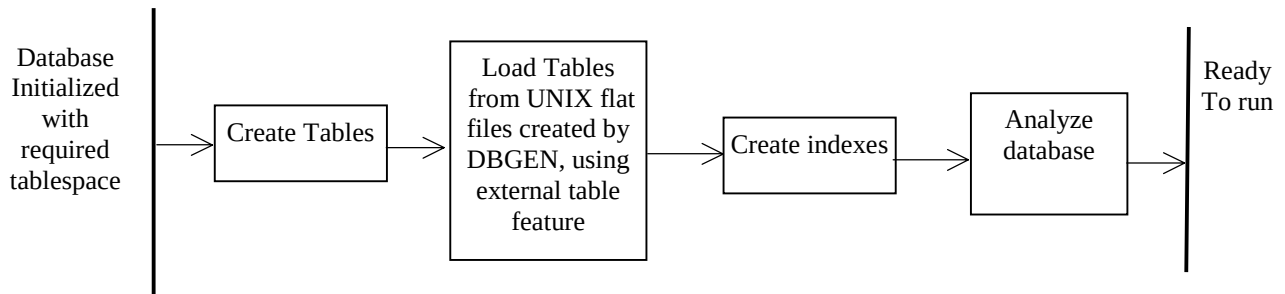
The data storage ration is computed from the following information:

Disk Type	# of Disks	Space per Disk (GB)	Sub-Total Disk Space
18.2GB/OK RPM Ultra3-SCSI	330	16.96	5,596.8GB
9.1 GB 7200 RPM Ultra-SCSI	6	8.48	50.9GB
		Total Space =	5,647.7GB
		Data Storage Ration =	56.48

5.9 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.

All load scripts are included in Appendix B.



5.10 Qualification Database Configuration

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

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

6 Clause 5: Performance Metrics and Execution Rules

6.1 System Activity Between Load and Performance Tests

Any system activity on the SUT that takes place between the conclusion of the load test and the beginning of the performance test must be fully disclosed including listings of scripts or command logs.

1. Executed DBTABLES.SQL using sqlplus
2. Executed FIRSTTEN.SQL using sqlplus
3. Executed CHECKIDX.SQL using sqlplus

The scripts may be found in Appendix F.

6.2 Steps in the Power Test

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

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

6.3 Timing Intervals

The timing intervals (see Clause 5.3.7) for each query and for both refresh functions must be reported for the power test.

The timing intervals for each query and for both refresh functions are listed in the Numerical Quantities Summary earlier in this document (STREAM 00).

6.4 Number of Streams for the Throughput Test

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

Five (5) query streams were used for the throughput test in this benchmark.

6.5 Start/Finish Time of Each Query Stream

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

The start and finish time for each query stream are listed in the Numerical Quantities Summary earlier in this document.

6.6 Total Elapsed Time

The total elapsed time of the measurement interval (see Clause 5.3.6) must be reported for the throughput test.

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

6.7 Start/Finish Time for the Refresh Function

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

The start and finish times for each refresh function in the refresh stream is listed in the Numerical Quantities Summary earlier in this document.

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

The timing intervals (see Clause 5.3.7) for each query of each stream and for each refresh function must be reported for the throughput test.

The timing intervals for each query of each stream and for each refresh function is listed in the Numerical Quantities Summary earlier in this document..

6.9 Performance Metrics

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

The computed performance metric, related numerical quantities and the price performance metric is listed in the Numerical Quantities Summary earlier in this document.

6.10 Performance Metric and Numerical Quantities from Both Runs

The performance metric (QphH@Size) and the numerical quantities (TPC-H Power@Size and TPC-H Throughput@Size) from both of the runs must be disclosed (see Clause 5.4.1).

Run ID	QppH@100GB	QthH@100GB	QphH@100GB
Run 1	6581.5	4814.0	5628.8
Run 2	6596.2	4717.7	5578.4
%Difference	.00200	.02006	.00908

6.11 System Activity Between Performance Runs

Any activity on the SUT that takes place between the conclusion of Run1 and the beginning of Run2 must be fully disclosed including listings of scripts or command logs along with any system reboots or database restarts.

There was no activity between Run 1 and Run 2.

7 Clause 6: SUT and Driver Implementation

7.1 Driver

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

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

Following the Power Test, QGEN is again called with the subsequent 5 stream ids to generate new QET for each Throughput Test. Qexecpl.c is called to run all 5 streams simultaneously to execute the queries as above. Immediately after that, the TPCHus script is called to run all 5 update pairs to finish the throughput run.

7.2 Implementation-Specific Layer

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

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

The ISL program (qexecpl.c) utilizes the Oracle Call Interface (OCI) to communicate with the Oracle database on the SUT. EQTs directly generated by QGEN are read and submitted to the SUT via the ISL program (qexecpl.c) as dynamic SQL statements. The ISL program then fetches the query execution output and reports it to the user. Timings are taken at intervals specified in Section 5.3.7 of the TPC-H benchmark specification.

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

7.3 Profile-Directed Optimization

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

Profile-directed optimization was not used.

8 Clause 7: Pricing

8.1 Hardware and Software Used

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

A detailed list of hardware and software used in the priced system is listed in the Pricing Summary earlier in this document.

The Oracle software pricing contact is:

MaryBeth Pierantoni of Oracle Corporation
Email Address: mary.beth.pierantoni@oracle.com
Phone Number: 650-506-2118

8.2 Total Three Year Pricing

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.

See the Pricing Summary earlier in this document.

8.3 Availability Date

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

Category	Available Date
Server Hardware	July 15, 2002
Storage	July 15, 2002
Software	July 15, 2002

9 Clause 9: Audit Related Items

9.1 Auditor's Report & Attestation Letter

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



Benchmark Sponsor: Dave Stanley
Compaq Computer
110 Spit Brook Road
ZKO2-3/M31
Nashua NH, 03062

July 12, 2002

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

Platform: **HP AlphaServer ES45 Cluster**
Database Manager: **Oracle9i Release 2 with Real Application Clusters**
Operating System: **HP Tru64 UNIX V5.1A/IPK**

The results were:

CPU (Speed)	Memory	Disks	QphH@100GB
HP AlphaServer ES45 (4 nodes)			
Per node : 4 Alpha EV 68 (1000 MHz)	Per node : 8 MB Cache/cpu 32 GB Main	297 x 18.2 GB 5 x 9.1 GB ext.	5,578.4

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

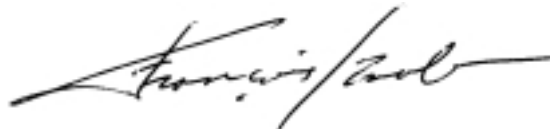
- The database records were defined with the proper layout and size

- The database population was generated using DBGEN
- The database was properly scaled to 100GB and populated accordingly
- The compliance of the database auxiliary data structures was verified
- The database load time was correctly measured and reported
- The required ACID properties were verified and met
- The query input variables were generated by QGEN
- The query text was produced using minor modifications
- The execution of the queries against the SF1 database produced compliant answers
- A compliant implementation specific layer was used to drive the tests
- The throughput tests involved 5 query streams
- The ratio between the longest and the shortest query was such that no query timing was adjusted
- The execution times for queries and refresh functions were correctly measured and reported
- The repeatability of the measured results was verified
- The required amount of database log was configured
- The system pricing was verified for major components and maintenance
- The major pages from the FDR were verified for accuracy

Additional Audit Notes:

None.

Respectfully Yours,

A handwritten signature in black ink, appearing to read 'François Raab', with a long horizontal flourish extending to the right.

François Raab
President

Appendix A: Tru64 UNIX and Oracle9i Parameters

Oracle9i Parameters

common.ora

```
cluster_database           = true
parallel_automatic_tuning = true
log_parallelism            = 2
audit_trail                = FALSE
compatible                 = 9.2.0
control_files               = (/usr/users/oracle/control_files/cntrl100gb_920_1.dbf,
$SF/control_files/cntrl100gb_920_2.dbf)
db_block_checksum          = false
db_block_size              = 32768
db_cache_size              = 5g
db_file_multiblock_read_count = 32
db_files                   = 2048
db_name                    = tpch
db_writer_processes        = 4
dml_locks                  = 20480
enqueue_resources          = 20480
global_names               = FALSE
java_pool_size             = 0
large_pool_size            = 3g
log_buffer                 = 100000
log_checkpoints_to_alert   = TRUE
max_rollback_segments      = 580
nls_date_format            = YYYY-MM-DD
open_cursors               = 1024
optimizer_features_enable  = 9.2.0.1
optimizer_index_cost_adj   = 25
optimizer_mode              = CHOOSE
parallel_adaptive_multi_user = FALSE
parallel_execution_message_size = 32768
parallel_max_servers       = 112
parallel_min_servers       = 112
parallel_threads_per_cpu   = 2
partition_view_enabled     = TRUE
processes                  = 1024
query_rewrite_enabled      = TRUE
replication_dependency_tracking = FALSE
sessions                   = 1024
shared_pool_size           = 1g
sort_area_size             = 24000000
hash_area_size             = 24000000
transaction_auditing       = FALSE
timed_statistics           = FALSE
trace_enabled              = FALSE
undo_management            = auto
max_commit_propagation_delay = 700
```

dbcreate.ora

```
# Init.ora for creating the 100GB TPC-H Oracle database
#
db_name                    = tpch
control_files              = (/usr/users/oracle/control_files/cntrl100gb_920_1.dbf,
$SF/control_files/cntrl100gb_920_2.dbf)
cluster_database           = FALSE
audit_trail                = FALSE
compatible                 = 9.2.0
db_block_buffers           = 2000
db_block_size              = 32768
db_block_checksum          = FALSE
db_files                   = 1024
db_file_multiblock_read_count = 16
global_names               = FALSE
log_checkpoints_to_alert   = true
log_checkpoint_interval    = 0
max_dump_file_size         = 5000
```

```
max_rollback_segments      = 160
nls_date_format            = YYYY-MM-DD
open_cursors               = 1024
optimizer_mode              = CHOOSE
processes                  = 256
query_rewrite_enabled      = true
sessions                   = 256
shared_pool_size           = 400000000
sort_area_size             = 1097152
sort_area_retained_size    = 1097152
transactions               = 512
undo_management            = auto
```

tpchorc1.ora (Node 1)

```
# init.ora for 300gb for Node 1 (tpchorc1)
#
instance_name=TPCH1
instance_number=1
parallel_instance_group=groupa
instance_groups=groupa,groupb
thread=1
ifile=$SF/control_files/common.ora
undo_tablespace=ts_undo1
```

tpchorc2.ora (Node 2)

```
# init.ora for 300gb for Node 2 (tpchorc2)
#
instance_name=TPCH2
instance_number=2
instance_groups=groupa
thread=2
ifile=$SF/control_files/common.ora
undo_tablespace=ts_undo2
```

tpchorc3.ora (Node 3)

```
# init.ora for 300gb for Node 3 (tpchorc3)
#
instance_name=TPCH3
instance_number=3
instance_groups=groupa
thread=3
ifile=$SF/control_files/common.ora
undo_tablespace=ts_undo3
```

tpchorc4.ora (Node 4)

```
# init.ora for 300gb for Node 4 (tpchorc4)
#
instance_name=TPCH4
instance_number=4
instance_groups=groupa
thread=4
ifile=$SF/control_files/common.ora
undo_tablespace=ts_undo4
```

Oracle9i Environment Variables

tpch.env.oracle

```
#!/usr/bin/csh
set path=(. $HOME $HOME/bin /sbin /usr/sbin /Oracle9i/app/product/9.2.0.0212/bin
/usr/bin/X11 $path /usr/local/bin /Oracle9i/bin /Oracle9i/app/product/9.2.0.0212 )
setenv TPCW_HOME /Oracle9i/app/tpcw
setenv BENCH_HOME /Oracle9i/bench/tpc
```

```

setenv BENCH_GEN $BENCH_HOME/tpcc
setenv GEN_SQL $BENCH_GEN/sql
setenv TPCC_SOURCE $BENCH_HOME/tpcc/source
setenv TPCC_SQL $BENCH_HOME/tpcc/sql
setenv TPCC_BLOCKS $BENCH_HOME/tpcc/blocks
setenv TPCC_SCRIPTS $BENCH_HOME/tpcc/scripts
setenv TPCC_UTILS $TPCC_SCRIPTS/utills
setenv AUDIT_SQL $BENCH_HOME/tpcc/audit/sql
setenv BUILD_SQL sql
setenv OUTDIR outdir
setenv MULT 1200
setenv SCRIPTS $BENCH_HOME/tpcw/scripts
setenv julier_scripts /Oracle9i/app/tpcw/scripts
setenv tool
#
setenv ORACLE_HOME /Oracle9i/app/product/9.2.0.0212
setenv ORACLE_BASE /Oracle9i/app/product/9.2.0.0212
setenv DBS $ORACLE_HOME/dbs
setenv ORACLE_817_HOME /Oracle/app/product/8.1.7
#
setenv ORA_NLS $ORACLE_HOME/ocommon/nls/admin/data
setenv ORA_NLS32 $ORACLE_HOME/ocommon/nls/admin/data
setenv ORACLE_TERM vt100
#
setenv ORACLE_SID TPCCH`clu_get_info | grep "this member" | awk '{print $6}`
#setenv ORACLE_SID valid
setenv SRCHOME $ORACLE_HOME
setenv ALERT_LOG_DIR $ORACLE_HOME/rdbms/log
setenv ALERT_LOG $ORACLE_HOME/rdbms/log/alert_$ORACLE_SID.log
unlimit stacksize
setenv O $ORACLE_HOME
#setenv PATH
$O/bin:$BENCH_HOME/util:$BENCH_HOME/benchrun/bin:$BENCH_HOME/bu
mpx:$BENCH_HOME/source:$PATH
#
##### tpchr Kit variables #####
#
setenv KIT_DIR $ORACLE_HOME/tpch
setenv SF $KIT_DIR/SF_100GB
setenv SFDIR $KIT_DIR/SF_100GB
setenv SFN 100
setenv SCALE_FACTOR 100
setenv DATABASE_USER tpcd/tpcd
setenv QUAL_DIR $KIT_DIR/SF_1GB
setenv DBNAME tpch
#
# variables that depend on Scale Factor
#
setenv UPDATE_DIR $KIT_DIR/update
setenv RES_DIR $KIT_DIR/queries_run/results
setenv SCHEMA_DIR $$SF/schema_load
setenv LOGS $$SF/logs
setenv BENCH_RUN $$SF/benchmark_run
setenv PIPELOAD $SCHEMA_DIR/pipe_load
setenv NUM_STREAMS 5
#
setenv PERL /usr/bin/perl
setenv BUMPX_DIR $KIT_DIR/bumpx
setenv BUMPX_OUT $KIT_DIR/bumpx
setenv UTILS $KIT_DIR/utills
setenv TEST_DB $tmp
setenv QUAL_DB $TEST_DB
setenv DBGENDIR $KIT_DIR/dbgen
setenv DBGEN $KIT_DIR/dbgen
setenv ACID_DIR $KIT_DIR/acid
setenv QEXEC $KIT_DIR/utills
setenv QUERIES $KIT_DIR/queries
setenv ANSWERS $KIT_DIR/answers
setenv ANS2VAL /tmp
setenv ACID_OUT $QUAL_DB/acid_out
setenv AUDIT $KIT_DIR/audit
setenv DSS_CONFIG $DBGENDIR
setenv DSS_QUERY $KIT_DIR/queries
setenv MAINT $KIT_DIR/maintenance
setenv CC cc
setenv FRAME $KIT_DIR/frame
setenv TEST $ORACLE_HOME/tpch/test
setenv LINKS $$SF/links
#setenv REGR_TEST $KIT_DIR/internal/regression_test

```

```

#
##### FRAME STUFF
#
setenv FRAME_PATH $KIT_DIR/frame
setenv FRAME_USER tpcd/tpcd
setenv RUN_PATH ""
#setenv ORACORE3INCL /vobs/oracore3/include
#setenv ORACORE3PUBL /vobs/oracore3/public
setenv ORACORE3INCL $ORACLE_HOME/rdbms/demo
setenv ORACORE3PUBL $ORACLE_HOME/rdbms/public
#setenv RDBMSPUBL /vobs/rdbms/public
setenv RDBMSPUBL $ORACLE_HOME/rdbms/public
#setenv NETWORKPUBL /vobs/network_src/public
setenv NETWORKPUBL $ORACLE_HOME/network/public
setenv RDBMSDEMO $ORACLE_HOME/rdbms/demo
setenv PLSQLEDEMO $ORACLE_HOME/plsql/demo
setenv PLSQLPUBL $ORACLE_HOME/plsql/public
setenv DSS_PATH /flatfiles
#setenv DSS_PATH /usr/pipe
#setenv PATH
./:${BUMPX_DIR}:${UTILS}:${DBGEN}:${MAINT}:${ACID_DIR}:${FRAME}/
bin:${FRAME}/bin:${REGR_TEST}:${PATH}
setenv PATH
./:${BUMPX_DIR}:${UTILS}:${DBGEN}:${MAINT}:${ACID_DIR}:${FRAME}/
bin:${FRAME}/bin:${PATH}
#
##### ENVIRONMENT VARIABLES #####
#
setenv WORKLOAD TPCCH
setenv HOST tpchorc1
#setenv INITORA $KIT_DIR/schema/test_db/testdb.ora
setenv INITORA $KIT_DIR/schema/test_db/sf100.ora
#
##### ALIASES #####
#

```

TRU64 UNIX Parameters

sysconfig_tpchorc1 (Node 1)

```

generic:
memberid=1
msgbuf_size=1048576
new_vers_high=1445664276479072064
new_vers_low=51969
act_vers_high=1445664276479072064
act_vers_low=51969
rolls_ver_lookup=0
    version_vendor = Compaq Computer Corporation
    version_avendor = COMPAQ
    version_product = Tru64 UNIX
    version_banner = Compaq Tru64 UNIX
locktype = 0
replicate_text = 1
dump_user_pte_pages = 1
dump_kernel_text = 1
live_dump_zero_suppress = 0

vm:
swapdevice=/dev/disk/dsk58b
vm_page_free_reserved=20
vm_page_free_min=30
    vm_swap_eager=0
    dump_user_pte_pages = 1
    new_wire_method = 1
    replicate_user_text = 1
    ubc_minpercent=5
    ubc_maxpercent=10
gh_chunks = 2750

aud97:
    Subsystem_Description = Creative Ensoniq Audio
    Device_Major_Req = Same
    Device_Char_Major = ANY
    Device_Char_Files = aud97
    Device_Char_Minor = 0

```



```

Module_Config_Name = aud97
AUD97_Developer_Debug = 0
Num_Installed = 1
PCI_Option = PCI_SE_Rev - 0x210, Vendor_Id - 0x1274, Device_Id -
0x1371, Rev - 0, Base - 0, Sub - 0, Pif - 0 Sub_Vid - 0, Sub_Did - 0, Vid_Mo_Flag -
1, Did_Mo_Flag - 1, Rev_Mo_Flag - 0, Base_Mo_Flag - 0, Sub_Mo_Flag - 0,
Pif_Mo_Flag - 0, Sub_Vid_Mo_Flag - 0, Sub_Did_Mo_Flag - 0, Driver_Name -
aud97, Type - C, Adpt_Config - N
CMA_Option = Size - 0x40000, Alignment - 0x10000, Addrlimit - 0,
Type - 29, Flag - 1

```

mmsess:

```

Subsystem_Description = ESS Tech. 1888/1887 audio driver
Device_Major_Req = Same
Device_Char_Major = ANY
Device_Char_Files = mmsess0
Device_Char_Minor = 0
Module_Config_Name = mmsess
Mmsess_Developer_Debug = 0
Num_Installed = 1
ISA_Option = Board_Id - Null, Function_Name - 'ES1888', Driver_Name -
mmsess, Type - C, Adpt_Config - N
EISA_Option = Board_Id - ISA2100, Function_Name - Null, Driver_Name -
mmsess, Type - C, Adpt_Config - N
CMA_Option = Size - 0x40000, Alignment - 0x10000, Addrlimit - 0,
Type - 29, Flag - 1
Playback_DMA_Channel_Override = -1
Playback_Mixer_Not_Used = 0
Playback_Mixer_Wave_Out = 0xff
Playback_Mixer_Microphone = 0
Playback_Mixer_Line_In = 0
Playback_Mixer_Synthesizer = 0
Playback_Mixer_Aux_CD = 255
Playback_Mixer_AuxB = 0
Record_Mixer_Wave_Out_1887 = 0
Record_Mixer_Microphone = 0
Record_Mixer_Line_In = 0
Record_Mixer_Synthesizer_1887 = 0
Record_Mixer_Aux_CD = 255
Record_Mixer_AuxB = 0
Pc_Speaker = 1
Input_Source = MIC
Record_Volume = 12
Master_Volume_Left = 0x20
Master_Volume_Right = 0x20
Microphone_Preamplifier_Enable = 0

```

per:

```

Module_Config_Name = per
PER_Developer_Debug = 0
PCI_Option = PCI_SE_Rev - 0x210, Vendor_Id - 0x10ba, Device_Id -
0x304, Vid_Mo_Flag - 1, Did_Mo_Flag - 1, Rev_Mo_Flag - 0, Base_Mo_Flag - 0,
Sub_Mo_Flag - 0, Pif_Mo_Flag - 0, Sub_Vid_Mo_Flag - 0, Sub_Did_Mo_Flag - 0,
Driver_Name - per, Type - C, Adpt_Config - N

```

rdg:

```

max_objs=8960
max_async_req=1000
msg_size=32768

```

inet:

```

udp_rcvspace=65535
udp_sendspace=65535

```

lsm:

```

lsm_rootdev_is_volume=0
Enable_LSM_Stats = 1

```

rt:

```

aio_task_max_num = 2048

```

io:

```

basic_dma_window_size = 2048

```

proc:

```

give_boost = 0
max_per_proc_stack_size = 33554432
max_per_proc_address_space = 128000000000
max_per_proc_data_size = 128000000000

```

```

max_proc_per_user = 2048
max_threads_per_user = 2048
per_proc_address_space = 128000000000
per_proc_data_size = 128000000000

```

ipc:

```

msg_max = 32768
msg_mnb = 30000
msg_mni = 256
msg_tql = 1024
sem_aem = 32768
sem_mni = 1024
sem_msl = 2000
sem_opm = 2000
sem_ume = 2000
sem_vmx = 320000
shm_max = 32212254720
shm_mni = 1024
shm_seg = 512
shm_threshold = 0
shm_allocate_striped = 1

```

sec:

```

acl_mode = disable

```

clubase:

```

cluster_expected_votes=1
cluster_name=tpchorch
cluster_node_name=tpchorc1
cluster_node_inter_name=tpchorc1-ics0
cluster_node_votes=1
cluster_interconnect=mct
cluster_seqdisk_major=19
cluster_seqdisk_minor=1016
cluster_qdisk_major=0
cluster_qdisk_minor=0
cluster_qdisk_votes=0

```

pcount:

```

Subsystem_Description = pcount device driver
Module_Config_Name = pcount
Module_Type = Dynamic

```

#

```

Device_Major_Req = Same
Device_Char_Major = ANY
Device_Char_Minor = 0
Device_Char_Files = pcount0

```

sysconfig_tpchorc2 (Node 2)

generic:

```

memberid=2
msgbuf_size=1048576
new_vers_high=1445664276479072064
new_vers_low=51969
act_vers_high=1445664276479072064
act_vers_low=51969
rolls_ver_lookup=0
version_vendor = Compaq Computer Corporation
version_avendor = COMPAQ
version_product = Tru64 UNIX
version_banner = Compaq Tru64 UNIX
locktype = 0
replicate_text = 1
dump_user_pte_pages = 1
dump_kernel_text = 1
live_dump_zero_suppress = 0

```

vm:

```

swapdevice=/dev/disk/dsk59b
vm_page_free_reserved=20
vm_page_free_min=30
vm_swap_eager=0
dump_user_pte_pages = 1
new_wire_method = 1
replicate_user_text = 1
ubc_minpercent=5
ubc_maxpercent=10
gh_chunks = 2750

```

```

rdg:
    max_objs=8960
    max_async_req=1000
    msg_size=32768

inet:
    udp_rcvspace=65535
    udp_sendspace=65535

lsm:
    lsm_rootdev_is_volume=0
    Enable_LSM_Stats = 1

clubase:
    cluster_expected_votes=1
    cluster_name=tpchorch
    cluster_node_name=tpchorc2
    cluster_node_inter_name=tpchorc2-ics0
    cluster_node_votes=0
    cluster_interconnect=mct
    cluster_seqdisk_major=19
    cluster_seqdisk_minor=1032
    cluster_qdisk_major=0
    cluster_qdisk_minor=0
    cluster_qdisk_votes=0

sec:
    acl_mode = disable

per:
    Module_Config_Name = per
    PER_Developer_Debug = 0
    PCI_Option = PCI_SE_Rev - 0x210, Vendor_Id - 0x10ba, Device_Id -
0x304, Vid_Mo_Flag - 1, Did_Mo_Flag - 1, Rev_Mo_Flag - 0, Base_Mo_Flag - 0,
Sub_Mo_Flag - 0, Pif_Mo_Flag - 0, Sub_Vid_Mo_Flag - 0, Sub_Did_Mo_Flag - 0,
Driver_Name - per, Type - C, Adpt_Config - N

rt:
    aio_task_max_num = 2048

io:
    basic_dma_window_size = 2048

proc:
    give_boost = 0
    max_per_proc_stack_size = 33554432
    max_per_proc_address_space = 128000000000
    max_per_proc_data_size = 128000000000
    max_proc_per_user = 2048
    max_threads_per_user = 2048
    per_proc_address_space = 128000000000
    per_proc_data_size = 128000000000

ipc:
    msg_max = 32768
    msg_mnb = 30000
    msg_mni = 256
    msg_tql = 1024
    sem_aem = 32768
    sem_mni = 1024
    sem_msl = 2000
    sem_opm = 2000
    sem_ume = 2000
    sem_vmx = 320000
    shm_max = 32212254720
    shm_mni = 1024
    shm_seg = 512
    ssm_threshold = 0
    shm_allocate_striped = 1

aud97:
    Subsystem_Description = Creative Ensoniq Audio
    Device_Major_Req = Same
    Device_Char_Major = ANY
    Device_Char_Files = aud97
    Device_Char_Minor = 0
    Module_Config_Name = aud97
    AUD97_Developer_Debug = 0

```

```

    Num_Installed = 1
    PCI_Option = PCI_SE_Rev - 0x210, Vendor_Id - 0x1274, Device_Id -
0x1371, Rev - 0, Base - 0, Sub - 0, Pif - 0 Sub_Vid - 0, Sub_Did - 0, Vid_Mo_Flag -
1, Did_Mo_Flag - 1, Rev_Mo_Flag - 0, Base_Mo_Flag - 0, Sub_Mo_Flag - 0,
Pif_Mo_Flag - 0, Sub_Vid_Mo_Flag - 0, Sub_Did_Mo_Flag - 0, Driver_Name -
aud97, Type - C, Adpt_Config - N
    CMA_Option = Size - 0x40000, Alignment - 0x10000, Addrlimit - 0,
Type - 29, Flag - 1

mmsess:
    Subsystem_Description = ESS Tech. 1888/1887 audio driver
    Device_Major_Req = Same
    Device_Char_Major = ANY
    Device_Char_Files = mmsess0
    Device_Char_Minor = 0
    Module_Config_Name = mmsess
    Mmsess_Developer_Debug = 0
    Num_Installed = 1
    ISA_Option = Board_Id - Null, Function_Name - 'ES1888', Driver_Name -
mmsess, Type - C, Adpt_Config - N
    EISA_Option = Board_Id - ISA2100, Function_Name - Null, Driver_Name -
mmsess, Type - C, Adpt_Config - N
    CMA_Option = Size - 0x40000, Alignment - 0x10000, Addrlimit - 0,
Type - 29, Flag - 1
    Playback_DMA_Channel_Override = -1
    Playback_Mixer_Not_Used = 0
    Playback_Mixer_Wave_Out = 0xff
    Playback_Mixer_Microphone = 0
    Playback_Mixer_Line_In = 0
    Playback_Mixer_Synthesizer = 0
    Playback_Mixer_Aux_CD = 255
    Playback_Mixer_AuxB = 0
    Record_Mixer_Wave_Out_1887 = 0
    Record_Mixer_Microphone = 0
    Record_Mixer_Line_In = 0
    Record_Mixer_Synthesizer_1887 = 0
    Record_Mixer_Aux_CD = 255
    Record_Mixer_AuxB = 0
    Pc_Speaker = 1
    Input_Source = MIC
    Record_Volume = 12
    Master_Volume_Left = 0x20
    Master_Volume_Right = 0x20
    Microphone_Preamplifier_Enable = 0

```

sysconfig_tpchorc3 (Node 3)

```

generic:
    memberid=3
    msgbuf_size=1048576
    new_vers_high=1445664276479072064
    new_vers_low=51969
    act_vers_high=1445664276479072064
    act_vers_low=51969
    rolls_ver_lookup=0
    version_vendor = Compaq Computer Corporation
    version_avendor = COMPAQ
    version_product = Tru64 UNIX
    version_banner = Compaq Tru64 UNIX
    locktype = 0
    replicate_text = 1
    dump_user_pte_pages = 1
    dump_kernel_text = 1
    live_dump_zero_suppress = 0

vm:
    swapdevice=/dev/disk/dsk60b
    vm_page_free_reserved=20
    vm_page_free_min=30
    vm_swap_eager=0
    dump_user_pte_pages = 1
    new_wire_method = 1
    replicate_user_text = 1
    ubc_minpercent=5
    ubc_maxpercent=10
    gh_chunks = 2750

rdg:

```

```

max_objs=8960
max_async_req=1000
msg_size=32768

inet:
    udp_rcvspace=65535
    udp_sendspace=65535

lsm:
    lsm_rootdev_is_volume=0
    Enable_LSM_Stats = 1

rt:
    aio_task_max_num = 2048

io:
    basic_dma_window_size = 2048

proc:
    give_boost = 0
    max_per_proc_stack_size = 33554432
    max_per_proc_address_space = 128000000000
    max_per_proc_data_size = 128000000000
    max_proc_per_user = 2048
    max_threads_per_user = 2048
    per_proc_address_space = 128000000000
    per_proc_data_size = 128000000000

ipc:
    msg_max = 32768
    msg_mnb = 30000
    msg_mni = 256
    msg_tql = 1024
    sem_aem = 32768
    sem_mni = 1024
    sem_msl = 2000
    sem_opm = 2000
    sem_ume = 2000
    sem_vmx = 320000
    shm_max = 32212254720
    shm_mni = 1024
    shm_seg = 512
    ssm_threshold = 0
    shm_allocate_striped = 1

clubase:
    cluster_expected_votes=1
    cluster_name=tpchorch
    cluster_node_name=tpchorc3
    cluster_node_inter_name=tpchorc3-ics0
    cluster_node_votes=0
    cluster_interconnect=mct
    cluster_seqdisk_major=19
    cluster_seqdisk_minor=1080
    cluster_qdisk_major=0
    cluster_qdisk_minor=0
    cluster_qdisk_votes=0

sec:
    acl_mode = disable

per:
    Module_Config_Name = per
    PER_Developer_Debug = 0
    PCI_Option = PCI_SE_Rev - 0x210, Vendor_Id - 0x10ba, Device_Id -
0x304, Vid_Mo_Flag - 1, Did_Mo_Flag - 1, Rev_Mo_Flag - 0, Base_Mo_Flag - 0,
Sub_Mo_Flag - 0, Pif_Mo_Flag - 0, Sub_Vid_Mo_Flag - 0, Sub_Did_Mo_Flag - 0,
Driver_Name - per, Type - C, Adpt_Config - N

aud97:
    Subsystem_Description = Creative Ensoniq Audio
    Device_Major_Req = Same
    Device_Char_Major = ANY
    Device_Char_Files = aud97
    Device_Char_Minor = 0
    Module_Config_Name = aud97
    AUD97_Developer_Debug = 0
    Num_Installed = 1

```

```

PCI_Option = PCI_SE_Rev - 0x210, Vendor_Id - 0x1274, Device_Id -
0x1371, Rev - 0, Base - 0, Sub - 0, Pif - 0 Sub_Vid - 0, Sub_Did - 0, Vid_Mo_Flag -
1, Did_Mo_Flag - 1, Rev_Mo_Flag - 0, Base_Mo_Flag - 0, Sub_Mo_Flag - 0,
Pif_Mo_Flag - 0, Sub_Vid_Mo_Flag - 0, Sub_Did_Mo_Flag - 0, Driver_Name -
aud97, Type - C, Adpt_Config - N
CMA_Option = Size - 0x40000, Alignment - 0x10000, Addrlimit - 0,
Type - 29, Flag - 1

mmsess:
    Subsystem_Description = ESS Tech. 1888/1887 audio driver
    Device_Major_Req = Same
    Device_Char_Major = ANY
    Device_Char_Files = mmsess0
    Device_Char_Minor = 0
    Module_Config_Name = mmsess
    Mmsess_Developer_Debug = 0
    Num_Installed = 1
    ISA_Option = Board_Id - Null, Function_Name - 'ES1888' , Driver_Name -
mmsess, Type - C, Adpt_Config - N
    EISA_Option = Board_Id - ISA2100, Function_Name - Null, Driver_Name -
mmsess, Type - C, Adpt_Config - N
    CMA_Option = Size - 0x40000, Alignment - 0x10000, Addrlimit - 0,
Type - 29, Flag - 1
    Playback_DMA_Channel_Override = -1
    Playback_Mixer_Not_Used = 0
    Playback_Mixer_Wave_Out = 0xff
    Playback_Mixer_Microphone = 0
    Playback_Mixer_Line_In = 0
    Playback_Mixer_Synthesizer = 0
    Playback_Mixer_Aux_CD = 255
    Playback_Mixer_AuxB = 0
    Record_Mixer_Wave_Out_1887 = 0
    Record_Mixer_Microphone = 0
    Record_Mixer_Line_In = 0
    Record_Mixer_Synthesizer_1887 = 0
    Record_Mixer_Aux_CD = 255
    Record_Mixer_AuxB = 0
    Pc_Speaker = 1
    Input_Source = MIC
    Record_Volume = 12
    Master_Volume_Left = 0x20
    Master_Volume_Right = 0x20
    Microphone_Preampl_Enable = 0

```

sysconfig_tpchorc4 (Node 4)

```

generic:
    memberid=4
    msgbuf_size=1048576
    new_vers_high=1445664276479072064
    new_vers_low=51969
    act_vers_high = 1445664276479072064
    act_vers_low=51969
    rolls_ver_lookup=0
    version_vendor = Compaq Computer Corporation
    version_avendor = COMPAQ
    version_product = Tru64 UNIX
    version_banner = Compaq Tru64 UNIX
    locktype = 0
    replicate_text = 1
    dump_user_pte_pages = 1
    dump_kernel_text = 1
    live_dump_zero_suppress = 0

vm:
    swapdevice=/dev/disk/dsk62b
    vm_page_free_reserved=20
    vm_page_free_min=30
    vm_swap_eager=0
    dump_user_pte_pages = 1
    new_wire_method = 1
    replicate_user_text = 1
    ubc_minpercent=5
    ubc_maxpercent=10
    gh_chunks = 2750

rdg:
    max_objs=8960

```

```

max_async_req=1000
msg_size=32768

inet:
  udp_rcvspace=65535
  udp_sendspace=65535

lsm:
  lsm_rootdev_is_volume=0
  Enable_LSM_Stats = 1

rt:
  aio_task_max_num = 2048

io:
  basic_dma_window_size = 2048

proc:
  give_boost = 0
  max_per_proc_stack_size = 33554432
  max_per_proc_address_space = 1280000000000
  max_per_proc_data_size = 1280000000000
  max_proc_per_user = 2048
  max_threads_per_user = 2048
  per_proc_address_space = 1280000000000
  per_proc_data_size = 1280000000000

ipc:
  msg_max = 32768
  msg_mnb = 30000
  msg_mni = 256
  msg_tql = 1024
  sem_aem = 32768
  sem_mni = 1024
  sem_msl = 2000
  sem_opm = 2000
  sem_ume = 2000
  sem_vmx = 320000
  shm_max = 32212254720
  shm_mni = 1024
  shm_seg = 512
  ssm_threshold = 0
  shm_allocate_striped = 1

clubase:
  cluster_expected_votes=1
  cluster_name=tpchorch
  cluster_node_name=tpchorc4
  cluster_node_inter_name=tpchorc4-ics0
  cluster_node_votes=0
  cluster_interconnect=mct
  cluster_seqdisk_major=19
  cluster_seqdisk_minor=1144
  cluster_qdisk_major=0
  cluster_qdisk_minor=0
  cluster_qdisk_votes=0

sec:
  acl_mode = disable

per:
  Module_Config_Name = per
  PER_Developer_Debug = 0
  PCI_Option = PCI_SE_Rev - 0x210, Vendor_Id - 0x10ba, Device_Id -
0x304, Vid_Mo_Flag - 1, Did_Mo_Flag - 1, Rev_Mo_Flag - 0, Base_Mo_Flag - 0,
Sub_Mo_Flag - 0, Pif_Mo_Flag - 0, Sub_Vid_Mo_Flag - 0, Sub_Did_Mo_Flag - 0,
Driver_Name - per, Type - C, Adpt_Config - N

aud97:
  Subsystem_Description = Creative Ensoniq Audio
  Device_Major_Req = Same
  Device_Char_Major = ANY
  Device_Char_Files = aud97
  Device_Char_Minor = 0
  Module_Config_Name = aud97
  AUD97_Developer_Debug = 0
  Num_Installed = 1
  PCI_Option = PCI_SE_Rev - 0x210, Vendor_Id - 0x1274, Device_Id -
0x1371, Rev - 0, Base - 0, Sub - 0, Pif - 0 Sub_Vid - 0, Sub_Did - 0, Vid_Mo_Flag -
1, Did_Mo_Flag - 1, Rev_Mo_Flag - 0, Base_Mo_Flag - 0, Sub_Mo_Flag - 0,
Pif_Mo_Flag - 0, Sub_Vid_Mo_Flag - 0, Sub_Did_Mo_Flag - 0, Driver_Name -
aud97, Type - C, Adpt_Config - N
  CMA_Option = Size - 0x40000, Alignment - 0x10000, Addrlimit - 0,
Type - 29, Flag - 1

mmsess:
  Subsystem_Description = ESS Tech. 1888/1887 audio driver
  Device_Major_Req = Same
  Device_Char_Major = ANY
  Device_Char_Files = mmsess0
  Device_Char_Minor = 0
  Module_Config_Name = mmsess
  Mmsess_Developer_Debug = 0
  Num_Installed = 1
  ISA_Option = Board_Id - Null, Function_Name - 'ES1888', Driver_Name -
mmsess, Type - C, Adpt_Config - N
  EISA_Option = Board_Id - ISA2100, Function_Name - Null, Driver_Name -
mmsess, Type - C, Adpt_Config - N
  CMA_Option = Size - 0x40000, Alignment - 0x10000, Addrlimit - 0,
Type - 29, Flag - 1
  Playback_DMA_Channel_Override = -1
  Playback_Mixer_Not_Used = 0
  Playback_Mixer_Wave_Out = 0xff
  Playback_Mixer_Microphone = 0
  Playback_Mixer_Line_In = 0
  Playback_Mixer_Synthesizer = 0
  Playback_Mixer_Aux_CD = 255
  Playback_Mixer_AuxB = 0
  Record_Mixer_Wave_Out_1887 = 0
  Record_Mixer_Microphone = 0
  Record_Mixer_Line_In = 0
  Record_Mixer_Synthesizer_1887 = 0
  Record_Mixer_Aux_CD = 255
  Record_Mixer_AuxB = 0
  Pc_Speaker = 1
  Input_Source = MIC
  Record_Volume = 12
  Master_Volume_Left = 0x20
  Master_Volume_Right = 0x20
  Microphone_Preamplifier_Enable = 0

```

Appendix B: Scripts and Programs

Database Create and Load

create_database_4node.sh

```
#!/bin/ksh
#
# Create 100GB Oracle TPC-H Database - Compaq Computer Corporation
# Performance Engineer: Eric L. Speed
#
date
sqlplus <<EOF
/ as sysdba
set echo on
spool create_database_4node_spool.log
startup nomount pfile= $SF/control_files/dbcreate.ora ;
create database tpch
controlfile reuse
logfile '$LINKS/redolog1' size 3303m reuse,
        '$LINKS/redolog2' size 3303m reuse,
        '$LINKS/redolog3' size 3303m reuse
datafile '$LINKS/system1' size 3303m reuse
undo tablespace ts_undo1 datafile '$SF/links/ts_undo1' size 3303m reuse
maxdatafiles 750
maxinstances 4;

alter tablespace ts_undo1 add datafile '$SF/links/ts_undo2' size 3303m reuse;

-- The next lines are added for the 2nd instance's redolog files

alter database add logfile thread 2 '$LINKS/redolog4' size 3303m reuse;
alter database add logfile thread 2 '$LINKS/redolog5' size 3303m reuse;
alter database add logfile thread 2 '$LINKS/redolog6' size 3303m reuse;

alter database add logfile thread 3 '$LINKS/redolog7' size 3303m reuse;
alter database add logfile thread 3 '$LINKS/redolog8' size 3303m reuse;
alter database add logfile thread 3 '$LINKS/redolog9' size 3303m reuse;

alter database add logfile thread 4 '$LINKS/redolog10' size 3303m reuse;
alter database add logfile thread 4 '$LINKS/redolog11' size 3303m reuse;
alter database add logfile thread 4 '$LINKS/redolog12' size 3303m reuse;

alter database enable public thread 2;
alter database enable public thread 3;
alter database enable public thread 4;

alter tablespace system add datafile '$SF/links/system2' size 3303m reuse;
alter tablespace system add datafile '$SF/links/system3' size 3303m reuse;
alter tablespace system add datafile '$SF/links/system4' size 3303m reuse;

create undo tablespace ts_undo2 datafile '$SF/links/ts_undo3' size 3303m reuse;
alter tablespace ts_undo2 add datafile '$SF/links/ts_undo4' size 3303m reuse;

create undo tablespace ts_undo3 datafile '$SF/links/ts_undo5' size 3303m reuse;
alter tablespace ts_undo3 add datafile '$SF/links/ts_undo6' size 3303m reuse;

create undo tablespace ts_undo4 datafile '$SF/links/ts_undo7' size 3303m reuse;
alter tablespace ts_undo4 add datafile '$SF/links/ts_undo8' size 3303m reuse;

-- Building data dictionary

set termout off
set echo off
@$ORACLE_HOME/rdbms/admin/catalog.sql;
@$ORACLE_HOME/rdbms/admin/catpar.sql;
@$ORACLE_HOME/rdbms/admin/catproc.sql;
exit;
EOF

sqlplus system/manager <<EOF
drop user tpch cascade;
```

```
grant DBA
to tpch identified by tpch;
```

```
@$ORACLE_HOME/rdbms/admin/utlxplan.sql;
@$ORACLE_HOME/sqlplus/admin/pupbld.sql;
exit;
EOF
```

```
sqlplus <<EOF
/ as sysdba
set echo on
set termout on
@$SF/schema_load/create_tablespace_schema.sql
@$SF/schema_load/create_tablespace_orders.sql
@$SF/schema_load/create_tablespace_lineitem.sql
exit;
EOF
```

```
sqlplus tpch/tpch <<EOF
@$SF/schema_load/create_tables_all_et.sql
@$ORACLE_HOME/rdbms/admin/utlxplan.sql;
exit;
EOF
```

```
sqlplus system/manager <<EOF
alter user tpch temporary tablespace ts_temp;
alter user tpch default tablespace ts_s;
spool off
exit;
EOF
date
```

create_tablespace_lineitem.sql

```
set echo on ;

-- tablespace ts_l1 thru ts_l12 are for the LineItem Partitions

drop tablespace ts_l1 including contents;
create tablespace ts_l1 datafile '$SF/links/ts_l1_1' size 10240m reuse
;
drop tablespace ts_l2 including contents;
create tablespace ts_l2 datafile '$SF/links/ts_l2_1' size 10240m reuse
;
drop tablespace ts_l3 including contents;
create tablespace ts_l3 datafile '$SF/links/ts_l3_1' size 10240m reuse
;
drop tablespace ts_l4 including contents;
create tablespace ts_l4 datafile '$SF/links/ts_l4_1' size 10240m reuse
;
drop tablespace ts_l5 including contents;
create tablespace ts_l5 datafile '$SF/links/ts_l5_1' size 10240m reuse
;
drop tablespace ts_l6 including contents;
create tablespace ts_l6 datafile '$SF/links/ts_l6_1' size 10240m reuse
;
drop tablespace ts_l7 including contents;
create tablespace ts_l7 datafile '$SF/links/ts_l7_1' size 10240m reuse
;
drop tablespace ts_l8 including contents;
create tablespace ts_l8 datafile '$SF/links/ts_l8_1' size 10240m reuse
;
drop tablespace ts_l9 including contents;
create tablespace ts_l9 datafile '$SF/links/ts_l9_1' size 10240m reuse
;
drop tablespace ts_l10 including contents;
create tablespace ts_l10 datafile '$SF/links/ts_l10_1' size 10240m reuse
;
drop tablespace ts_l11 including contents;
create tablespace ts_l11 datafile '$SF/links/ts_l11_1' size 10240m reuse
;
drop tablespace ts_l12 including contents;
create tablespace ts_l12 datafile '$SF/links/ts_l12_1' size 10240m reuse
;
alter tablespace ts_l1 add datafile '$SF/links/ts_l1_2' size 10240m reuse;
alter tablespace ts_l2 add datafile '$SF/links/ts_l2_2' size 10240m reuse;
alter tablespace ts_l3 add datafile '$SF/links/ts_l3_2' size 10240m reuse;
alter tablespace ts_l4 add datafile '$SF/links/ts_l4_2' size 10240m reuse;
```

```

alter tablespace ts_l5 add datafile '$SF/links/ts_l5_2' size 10240m reuse;
alter tablespace ts_l6 add datafile '$SF/links/ts_l6_2' size 10240m reuse;
alter tablespace ts_l7 add datafile '$SF/links/ts_l7_2' size 10240m reuse;
alter tablespace ts_l8 add datafile '$SF/links/ts_l8_2' size 10240m reuse;
alter tablespace ts_l9 add datafile '$SF/links/ts_l9_2' size 10240m reuse;
alter tablespace ts_l10 add datafile '$SF/links/ts_l10_2' size 10240m reuse;
alter tablespace ts_l11 add datafile '$SF/links/ts_l11_2' size 10240m reuse;
alter tablespace ts_l12 add datafile '$SF/links/ts_l12_2' size 10240m reuse;

```

create_tablespace_orders.sql

```

set echo on ;
-- tablespace ts_o1 thru ts_o12 are for the Orders Partitions

drop tablespace ts_o1 including contents;
create tablespace ts_o1 datafile '$SF/links/ts_o1_1' size 2400m reuse;
drop tablespace ts_o2 including contents;
create tablespace ts_o2 datafile '$SF/links/ts_o2_1' size 2400m reuse;
drop tablespace ts_o3 including contents;
create tablespace ts_o3 datafile '$SF/links/ts_o3_1' size 2400m reuse;
drop tablespace ts_o4 including contents;
create tablespace ts_o4 datafile '$SF/links/ts_o4_1' size 2400m reuse;
drop tablespace ts_o5 including contents;
create tablespace ts_o5 datafile '$SF/links/ts_o5_1' size 2400m reuse;
drop tablespace ts_o6 including contents;
create tablespace ts_o6 datafile '$SF/links/ts_o6_1' size 2400m reuse;
drop tablespace ts_o7 including contents;
create tablespace ts_o7 datafile '$SF/links/ts_o7_1' size 2400m reuse;
drop tablespace ts_o8 including contents;
create tablespace ts_o8 datafile '$SF/links/ts_o8_1' size 2400m reuse;
drop tablespace ts_o9 including contents;
create tablespace ts_o9 datafile '$SF/links/ts_o9_1' size 2400m reuse;
drop tablespace ts_o10 including contents;
create tablespace ts_o10 datafile '$SF/links/ts_o10_1' size 2400m reuse;
drop tablespace ts_o11 including contents;
create tablespace ts_o11 datafile '$SF/links/ts_o11_1' size 2400m reuse;
drop tablespace ts_o12 including contents;
create tablespace ts_o12 datafile '$SF/links/ts_o12_1' size 2400m reuse;
alter tablespace ts_o1 add datafile '$SF/links/ts_o1_2' size 2400m reuse;
alter tablespace ts_o2 add datafile '$SF/links/ts_o2_2' size 2400m reuse;
alter tablespace ts_o3 add datafile '$SF/links/ts_o3_2' size 2400m reuse;
alter tablespace ts_o4 add datafile '$SF/links/ts_o4_2' size 2400m reuse;
alter tablespace ts_o5 add datafile '$SF/links/ts_o5_2' size 2400m reuse;
alter tablespace ts_o6 add datafile '$SF/links/ts_o6_2' size 2400m reuse;
alter tablespace ts_o7 add datafile '$SF/links/ts_o7_2' size 2400m reuse;
alter tablespace ts_o8 add datafile '$SF/links/ts_o8_2' size 2400m reuse;
alter tablespace ts_o9 add datafile '$SF/links/ts_o9_2' size 2400m reuse;
alter tablespace ts_o10 add datafile '$SF/links/ts_o10_2' size 2400m reuse;
alter tablespace ts_o11 add datafile '$SF/links/ts_o11_2' size 2400m reuse;
alter tablespace ts_o12 add datafile '$SF/links/ts_o12_2' size 2400m reuse;

```

create_tablespace_schema.sql

```

--
-- Schema Tablespace Creation Phase
-- 100GB Schema
-- Edit History
--
-- ELS      24-May-2002 Create.
--
set echo on ;

-- Creating tablespaces

drop tablespace ts_s including contents;
create tablespace ts_s
    datafile '$SF/links/ts_s1' size 5120m reuse
    extent management local
    autoallocate;
alter tablespace ts_s add datafile '$SF/links/ts_s2' size 5120m reuse;

drop tablespace ts_c including contents;
create tablespace ts_c
    datafile '$SF/links/ts_c1' size 5120m reuse
    extent management local
    autoallocate;
alter tablespace ts_c add datafile '$SF/links/ts_c2' size 5120m reuse;

```

```

drop tablespace ts_ps including contents;
create tablespace ts_ps
    datafile '$SF/links/ts_ps1' size 5120m reuse
    extent management local
    autoallocate;

```

```

alter tablespace ts_ps add datafile '$SF/links/ts_ps2' size 5120m reuse;
alter tablespace ts_ps add datafile '$SF/links/ts_ps3' size 5120m reuse;
alter tablespace ts_ps add datafile '$SF/links/ts_ps4' size 5120m reuse;
alter tablespace ts_ps add datafile '$SF/links/ts_ps5' size 5120m reuse;
alter tablespace ts_ps add datafile '$SF/links/ts_ps6' size 5120m reuse;
alter tablespace ts_ps add datafile '$SF/links/ts_ps7' size 5120m reuse;
alter tablespace ts_ps add datafile '$SF/links/ts_ps8' size 5120m reuse;

```

```

drop tablespace ts_p including contents;
create tablespace ts_p
    datafile '$SF/links/ts_p1' size 5120m reuse
    extent management local
    autoallocate;
alter tablespace ts_p add datafile '$SF/links/ts_p2' size 5120m reuse;
alter tablespace ts_p add datafile '$SF/links/ts_p3' size 5120m reuse;
alter tablespace ts_p add datafile '$SF/links/ts_p4' size 5120m reuse;

```

```

drop tablespace ts_i_l including contents;
create tablespace ts_i_l
    datafile '$SF/links/ts_i_l1' size 5120m reuse
    extent management local
    autoallocate;
alter tablespace ts_i_l add datafile '$SF/links/ts_i_l2' size 5120m reuse;
alter tablespace ts_i_l add datafile '$SF/links/ts_i_l3' size 5120m reuse;
alter tablespace ts_i_l add datafile '$SF/links/ts_i_l4' size 5120m reuse;

```

-- creating tpcd's ts_i_o tablespace

```

drop tablespace ts_i_o including contents;
create tablespace ts_i_o
    datafile '$SF/links/ts_i_o1' size 5120m reuse
    extent management local
    autoallocate;
alter tablespace ts_i_o add datafile '$SF/links/ts_i_o2' size 5120m reuse;
alter tablespace ts_i_o add datafile '$SF/links/ts_i_o3' size 5120m reuse;
alter tablespace ts_i_o add datafile '$SF/links/ts_i_o4' size 5120m reuse;

```

```

drop tablespace ts_i_ps including contents;
create tablespace ts_i_ps
    datafile '$SF/links/ts_i_ps1' size 5120m reuse
    extent management local
    autoallocate;
alter tablespace ts_i_ps add datafile '$SF/links/ts_i_ps2' size 5120m reuse;
alter tablespace ts_i_ps add datafile '$SF/links/ts_i_ps3' size 5120m reuse;
alter tablespace ts_i_ps add datafile '$SF/links/ts_i_ps4' size 5120m reuse;

```

```

drop tablespace ts_i_c including contents;
create tablespace ts_i_c
    datafile '$SF/links/ts_i_c1' size 5120m reuse
    extent management local
    autoallocate;
alter tablespace ts_i_c add datafile '$SF/links/ts_i_c2' size 5120m reuse;

```

```

drop tablespace ts_temp including contents;
create temporary tablespace ts_temp tempfile '$SF/links/ts_temp1' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp2' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp3' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp4' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp5' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp6' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp7' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp8' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp9' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp10' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp11' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp12' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp13' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp14' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp15' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp16' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp17' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp18' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp19' size 8192m reuse;

```

```

alter tablespace ts_temp add tempfile '$SF/links/ts_temp20' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp21' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp22' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp23' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp24' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp25' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp26' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp27' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp28' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp29' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp30' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp31' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp32' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp33' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp34' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp35' size 8192m reuse;
alter tablespace ts_temp add tempfile '$SF/links/ts_temp36' size 8192m reuse;

```

create_tables_all_et.sql

```

drop directory data_dir;
create directory data_dir as '/flatfiles';

```

```

drop table lineitem_et;
create table lineitem_et (
    l_shipdate      date ,
    l_orderkey       number ,
    l_discount       number ,
    l_extendedprice  number ,
    l_supplykey      number ,
    l_quantity       number ,
    l_returnflag     char(1) ,
    l_partkey        number ,
    l_linestatus     char(1) ,
    l_tax            number ,
    l_commitdate     date ,
    l_receiptdate    date ,
    l_shipmode       char(10) ,
    l_linenum        number ,
    l_shipinstruct   char(25) ,
    l_comment        varchar(44)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
    records delimited by newline
    nobadfile
    nologfile
    nodiscardfile
    fields terminated by '|'
    missing field values are null
)
location (
'lineitem.tbl.1','lineitem.tbl.2','lineitem.tbl.3',
'lineitem.tbl.4','lineitem.tbl.5','lineitem.tbl.6',
'lineitem.tbl.7','lineitem.tbl.8','lineitem.tbl.9',
'lineitem.tbl.10','lineitem.tbl.11','lineitem.tbl.12',
'lineitem.tbl.13','lineitem.tbl.14','lineitem.tbl.15',
'lineitem.tbl.16'
))reject limit unlimited;
alter table lineitem_et parallel;

```

```

drop table orders_et;
create table orders_et (
    o_orderdate      date ,
    o_orderkey       number ,
    o_custkey        number ,
    o_orderpriority   char(15) ,
    o_shippriority    number ,
    o_clerk           char(15) ,
    o_orderstatus     char(1) ,
    o_totalprice      number ,
    o_comment         varchar(79)
)
organization external (
type ORACLE_LOADER

```

```

default directory data_dir
access parameters
(
    records delimited by newline
    nobadfile
    nologfile
    nodiscardfile
    fields terminated by '|'
    missing field values are null
)
location (
'orders.tbl.1','orders.tbl.2','orders.tbl.3',
'orders.tbl.4','orders.tbl.5','orders.tbl.6',
'orders.tbl.7','orders.tbl.8','orders.tbl.9',
'orders.tbl.10','orders.tbl.11','orders.tbl.12',
'orders.tbl.13','orders.tbl.14','orders.tbl.15',
'orders.tbl.16'
))reject limit unlimited;
alter table orders_et parallel;

```

```

drop table part_et;
create table part_et (
    p_partkey        number ,
    p_type           varchar(25) ,
    p_size           number ,
    p_brand          char(10) ,
    p_name           varchar(55) ,
    p_container      char(10) ,
    p_mfgr           char(25) ,
    p_retailprice     number ,
    p_comment        varchar(23)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
    records delimited by newline
    nobadfile
    nologfile
    nodiscardfile
    fields terminated by '|'
    missing field values are null
)
location (
'part.tbl.1','part.tbl.2','part.tbl.3',
'part.tbl.4','part.tbl.5','part.tbl.6',
'part.tbl.7','part.tbl.8','part.tbl.9',
'part.tbl.10','part.tbl.11','part.tbl.12',
'part.tbl.13','part.tbl.14','part.tbl.15',
'part.tbl.16'
))reject limit unlimited;
alter table part_et parallel;

```

```

drop table partsupp_et;
create table partsupp_et (
    ps_partkey       number ,
    ps_suppkey        number ,
    ps_supplycost     number ,
    ps_availqty       number ,
    ps_comment        varchar(199)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
    records delimited by newline
    nobadfile
    nologfile
    nodiscardfile
    fields terminated by '|'
    missing field values are null
)
location (
'partsupp.tbl.1','partsupp.tbl.2','partsupp.tbl.3',
'partsupp.tbl.4','partsupp.tbl.5','partsupp.tbl.6',
'partsupp.tbl.7','partsupp.tbl.8','partsupp.tbl.9',

```

```
'partsupp.tbl.10','partsupp.tbl.11','partsupp.tbl.12',
'partsupp.tbl.13','partsupp.tbl.14','partsupp.tbl.15',
'partsupp.tbl.16'
))reject limit unlimited;
alter table partsupp_et parallel;
```

```
drop table supplier_et;
create table supplier_et (
  s_supplierkey      number ,
  s_nationkey         number ,
  s_comment           varchar(101) ,
  s_name              char(25) ,
  s_address            varchar(40) ,
  s_phone              char(15) ,
  s_acctbal            number
)
```

```
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
  records delimited by newline
  nobadfile
  nologfile
  nodiscardfile
  fields terminated by '|'
  missing field values are null
)
```

```
location (
'supplier.tbl.1','supplier.tbl.2','supplier.tbl.3',
'supplier.tbl.4','supplier.tbl.5','supplier.tbl.6',
'supplier.tbl.7','supplier.tbl.8','supplier.tbl.9',
'supplier.tbl.10','supplier.tbl.11','supplier.tbl.12',
'supplier.tbl.13','supplier.tbl.14','supplier.tbl.15',
'supplier.tbl.16'
))reject limit unlimited;
alter table supplier_et parallel;
```

```
drop table customer_et;
create table customer_et (
  c_custkey          number ,
  c_mktsegment        char(10) ,
  c_nationkey         number ,
  c_name              varchar(25) ,
  c_address            varchar(40) ,
  c_phone              char(15) ,
  c_acctbal            number ,
  c_comment            varchar(117)
)
```

```
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
  records delimited by newline
  nobadfile
  nologfile
  nodiscardfile
  fields terminated by '|'
  missing field values are null
)
```

```
location (
'customer.tbl.1','customer.tbl.2','customer.tbl.3',
'customer.tbl.4','customer.tbl.5','customer.tbl.6',
'customer.tbl.7','customer.tbl.8','customer.tbl.9',
'customer.tbl.10','customer.tbl.11','customer.tbl.12',
'customer.tbl.13','customer.tbl.14','customer.tbl.15',
'customer.tbl.16'
))reject limit unlimited;
alter table customer_et parallel;
```

```
drop table nation_et;
create table nation_et (
  n_nationkey      number ,
  n_name            char(25) ,
  n_regionkey       number ,
  n_comment         varchar(152)
)
```

```
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
  records delimited by newline
  nobadfile
  nologfile
  nodiscardfile
  fields terminated by '|'
  missing field values are null
)
```

```
location (
'nation.tbl'
))reject limit unlimited;
alter table nation_et parallel;
```

```
drop table region_et;
create table region_et (
  r_regionkey      number ,
  r_name            char(25) ,
  r_comment         varchar(152)
)
```

```
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
  records delimited by newline
  nobadfile
  nologfile
  nodiscardfile
  fields terminated by '|'
  missing field values are null
)
```

```
location (
'region.tbl'
))reject limit unlimited;
alter table region_et parallel;
```

dbgen_flatfiles.sh

```
#!/bin/ksh
```

```
#####
#####
# Data (Flat File) Generation Phase
#####
#####
```

```
# Lineitem Table Flatfile Generation
```

```
dbgen -f -T L -s 100 -C 16 -S 1 &
dbgen -f -T L -s 100 -C 16 -S 2 &
dbgen -f -T L -s 100 -C 16 -S 3 &
dbgen -f -T L -s 100 -C 16 -S 4 &
dbgen -f -T L -s 100 -C 16 -S 5 &
dbgen -f -T L -s 100 -C 16 -S 6 &
dbgen -f -T L -s 100 -C 16 -S 7 &
dbgen -f -T L -s 100 -C 16 -S 8 &
wait
```

```
dbgen -f -T L -s 100 -C 16 -S 9 &
dbgen -f -T L -s 100 -C 16 -S 10 &
dbgen -f -T L -s 100 -C 16 -S 11 &
dbgen -f -T L -s 100 -C 16 -S 12 &
dbgen -f -T L -s 100 -C 16 -S 13 &
dbgen -f -T L -s 100 -C 16 -S 14 &
dbgen -f -T L -s 100 -C 16 -S 15 &
dbgen -f -T L -s 100 -C 16 -S 16 &
wait
```

```
# Orders Table Flatfile Generation
```

```
dbgen -f -T O -s 100 -C 16 -S 1 &
dbgen -f -T O -s 100 -C 16 -S 2 &
dbgen -f -T O -s 100 -C 16 -S 3 &
```



```
dbgen -f -T O -s 100 -C 16 -S 4 &
dbgen -f -T O -s 100 -C 16 -S 5 &
dbgen -f -T O -s 100 -C 16 -S 6 &
dbgen -f -T O -s 100 -C 16 -S 7 &
dbgen -f -T O -s 100 -C 16 -S 8 &
wait
```

```
dbgen -f -T O -s 100 -C 16 -S 9 &
dbgen -f -T O -s 100 -C 16 -S 10 &
dbgen -f -T O -s 100 -C 16 -S 11 &
dbgen -f -T O -s 100 -C 16 -S 12 &
dbgen -f -T O -s 100 -C 16 -S 13 &
dbgen -f -T O -s 100 -C 16 -S 14 &
dbgen -f -T O -s 100 -C 16 -S 15 &
dbgen -f -T O -s 100 -C 16 -S 16 &
wait
```

Part Table Flatfile Generation

```
dbgen -f -T P -s 100 -C 16 -S 1 &
dbgen -f -T P -s 100 -C 16 -S 2 &
dbgen -f -T P -s 100 -C 16 -S 3 &
dbgen -f -T P -s 100 -C 16 -S 4 &
dbgen -f -T P -s 100 -C 16 -S 5 &
dbgen -f -T P -s 100 -C 16 -S 6 &
dbgen -f -T P -s 100 -C 16 -S 7 &
dbgen -f -T P -s 100 -C 16 -S 8 &
wait
```

```
dbgen -f -T P -s 100 -C 16 -S 9 &
dbgen -f -T P -s 100 -C 16 -S 10 &
dbgen -f -T P -s 100 -C 16 -S 11 &
dbgen -f -T P -s 100 -C 16 -S 12 &
dbgen -f -T P -s 100 -C 16 -S 13 &
dbgen -f -T P -s 100 -C 16 -S 14 &
dbgen -f -T P -s 100 -C 16 -S 15 &
dbgen -f -T P -s 100 -C 16 -S 16 &
wait
```

Partsupp Table Flatfile Generation

```
dbgen -f -T S -s 100 -C 16 -S 1
dbgen -f -T S -s 100 -C 16 -S 2
dbgen -f -T S -s 100 -C 16 -S 3
dbgen -f -T S -s 100 -C 16 -S 4
dbgen -f -T S -s 100 -C 16 -S 5
dbgen -f -T S -s 100 -C 16 -S 6
dbgen -f -T S -s 100 -C 16 -S 7
dbgen -f -T S -s 100 -C 16 -S 8
dbgen -f -T S -s 100 -C 16 -S 9
dbgen -f -T S -s 100 -C 16 -S 10
dbgen -f -T S -s 100 -C 16 -S 11
dbgen -f -T S -s 100 -C 16 -S 12
dbgen -f -T S -s 100 -C 16 -S 13
dbgen -f -T S -s 100 -C 16 -S 14
dbgen -f -T S -s 100 -C 16 -S 15
dbgen -f -T S -s 1300 -C 16 -S 16
```

Supplier Table Flatfile Generation

```
dbgen -f -T s -s 100 -C 16 -S 1
dbgen -f -T s -s 100 -C 16 -S 2
dbgen -f -T s -s 100 -C 16 -S 3
dbgen -f -T s -s 100 -C 16 -S 4
dbgen -f -T s -s 100 -C 16 -S 5
dbgen -f -T s -s 100 -C 16 -S 6
dbgen -f -T s -s 100 -C 16 -S 7
dbgen -f -T s -s 100 -C 16 -S 8
dbgen -f -T s -s 100 -C 16 -S 9
dbgen -f -T s -s 100 -C 16 -S 10
dbgen -f -T s -s 100 -C 16 -S 11
dbgen -f -T s -s 100 -C 16 -S 12
dbgen -f -T s -s 100 -C 16 -S 13
dbgen -f -T s -s 100 -C 16 -S 14
dbgen -f -T s -s 100 -C 16 -S 15
dbgen -f -T s -s 100 -C 16 -S 16
```

Customer Table Flatfile Generation

```
dbgen -f -T c -s 100 -C 16 -S 1
dbgen -f -T c -s 100 -C 16 -S 2
dbgen -f -T c -s 100 -C 16 -S 3
dbgen -f -T c -s 100 -C 16 -S 4
dbgen -f -T c -s 100 -C 16 -S 5
dbgen -f -T c -s 100 -C 16 -S 6
dbgen -f -T c -s 100 -C 16 -S 7
dbgen -f -T c -s 100 -C 16 -S 8
dbgen -f -T c -s 100 -C 16 -S 9
dbgen -f -T c -s 100 -C 16 -S 10
dbgen -f -T c -s 100 -C 16 -S 11
dbgen -f -T c -s 100 -C 16 -S 12
dbgen -f -T c -s 100 -C 16 -S 13
dbgen -f -T c -s 100 -C 16 -S 14
dbgen -f -T c -s 100 -C 16 -S 15
dbgen -f -T c -s 100 -C 16 -S 16
```

Nation and Region Table Flatfile Generation

```
dbgen -f -T n -s 100 -C 1 -S 1
dbgen -f -T r -s 100 -C 1 -S 1
```

echo FlatFile Generation Complete...

shutdown_instance.sh

```
#!/bin/ksh
```

```
sqlplus <<EOF
/ as sysdba
shutdown
exit
EOF
```

startup_4_instances.sh

```
#!/bin/ksh
```

echo "Starting Up Oracle on tpchorc1 Now..."

\$SF/scripts/startup_tpchorc1.sh

echo "...Oracle Startup Complete on tpchorc1"

sleep 3

echo "Starting Up Oracle on tpchorc2 Now..."

rsh tpchorc2 \$SF/scripts/startup_tpchorc2.sh

echo "...Oracle Startup Complete on tpchorc2"

sleep 3

echo "Starting Up Oracle on tpchorc3 Now..."

rsh tpchorc3 \$SF/scripts/startup_tpchorc3.sh

echo "...Oracle Startup Complete on tpchorc3"

sleep 3

echo "Starting Up Oracle on tpchorc4 Now..."

rsh tpchorc4 \$SF/scripts/startup_tpchorc4.sh

echo "...Oracle Startup Complete on tpchorc4"

startup_tpchorc1.sh

```
#!/bin/ksh
```

```
sqlplus <<EOF
/ as sysdba
startup parallel pfile=$SF/control_files/tpchorc1.ora
```

```
exit
EOF
```

startup_tpchorc2.sh

```
#!/bin/ksh

sqlplus <<EOF
/ as sysdba
startup parallel pfile=$SF/control_files/tpchorc2.ora
exit
EOF
```

startup_tpchorc3.sh

```
#!/bin/ksh

sqlplus <<EOF
/ as sysdba
startup parallel pfile=$SF/control_files/tpchorc3.ora
exit
EOF
```

startup_tpchorc4.sh

```
#!/bin/ksh

sqlplus <<EOF
/ as sysdba
startup parallel pfile=$SF/control_files/tpchorc4.ora
exit
EOF
```

database_load_tables_et.sh

```
#!/bin/ksh

# Create All TPC-H Schema Tables and Load....
```

```
sqlplus tpchd/tpcd <<EOF
spool database_load_tables_et.log
set echo on;
drop table lineitem;
create table lineitem(
    l_shipdate        ,
    l_orderkey        NOT NULL,
    l_discount        ,
    l_extendedprice   ,
    l_suppkey         NOT NULL,
    l_quantity        ,
    l_returnflag      ,
    l_partkey         NOT NULL,
    l_linestatus      ,
    l_tax             NOT NULL,
    l_commitdate      ,
    l_receiptdate     ,
    l_shipmode        ,
    l_linenumber      NOT NULL,
    l_shipinstruct    ,
    l_comment
)
pctfree 1
pctused 99
initrans 10
compress
parallel(degree 8 instances 4)
```

```
nologging
partition by range (l_shipdate)
subpartition by hash(l_partkey)
subpartitions 16
(
    partition item1 values less than (to_date('1992-01-01','YYYY-MM-DD'))
    tablespace ts_11,
    partition item2 values less than (to_date('1992-02-01','YYYY-MM-DD'))
    tablespace ts_12,
    partition item3 values less than (to_date('1992-03-01','YYYY-MM-DD'))
    tablespace ts_13,
    partition item4 values less than (to_date('1992-04-01','YYYY-MM-DD'))
    tablespace ts_14,
    partition item5 values less than (to_date('1992-05-01','YYYY-MM-DD'))
    tablespace ts_15,
    partition item6 values less than (to_date('1992-06-01','YYYY-MM-DD'))
    tablespace ts_16,
    partition item7 values less than (to_date('1992-07-01','YYYY-MM-DD'))
    tablespace ts_17,
    partition item8 values less than (to_date('1992-08-01','YYYY-MM-DD'))
    tablespace ts_18,
    partition item9 values less than (to_date('1992-09-01','YYYY-MM-DD'))
    tablespace ts_19,
    partition item10 values less than (to_date('1992-10-01','YYYY-MM-DD'))
    tablespace ts_110,
    partition item11 values less than (to_date('1992-11-01','YYYY-MM-DD'))
    tablespace ts_111,
    partition item12 values less than (to_date('1992-12-01','YYYY-MM-DD'))
    tablespace ts_112,
    partition item13 values less than (to_date('1993-01-01','YYYY-MM-DD'))
    tablespace ts_11,
    partition item14 values less than (to_date('1993-02-01','YYYY-MM-DD'))
    tablespace ts_12,
    partition item15 values less than (to_date('1993-03-01','YYYY-MM-DD'))
    tablespace ts_13,
    partition item16 values less than (to_date('1993-04-01','YYYY-MM-DD'))
    tablespace ts_14,
    partition item17 values less than (to_date('1993-05-01','YYYY-MM-DD'))
    tablespace ts_15,
    partition item18 values less than (to_date('1993-06-01','YYYY-MM-DD'))
    tablespace ts_16,
    partition item19 values less than (to_date('1993-07-01','YYYY-MM-DD'))
    tablespace ts_17,
    partition item20 values less than (to_date('1993-08-01','YYYY-MM-DD'))
    tablespace ts_18,
    partition item21 values less than (to_date('1993-09-01','YYYY-MM-DD'))
    tablespace ts_19,
    partition item22 values less than (to_date('1993-10-01','YYYY-MM-DD'))
    tablespace ts_110,
    partition item23 values less than (to_date('1993-11-01','YYYY-MM-DD'))
    tablespace ts_111,
    partition item24 values less than (to_date('1993-12-01','YYYY-MM-DD'))
    tablespace ts_112,
    partition item25 values less than (to_date('1994-01-01','YYYY-MM-DD'))
    tablespace ts_11,
    partition item26 values less than (to_date('1994-02-01','YYYY-MM-DD'))
    tablespace ts_12,
    partition item27 values less than (to_date('1994-03-01','YYYY-MM-DD'))
    tablespace ts_13,
    partition item28 values less than (to_date('1994-04-01','YYYY-MM-DD'))
    tablespace ts_14,
    partition item29 values less than (to_date('1994-05-01','YYYY-MM-DD'))
    tablespace ts_15,
    partition item30 values less than (to_date('1994-06-01','YYYY-MM-DD'))
    tablespace ts_16,
    partition item31 values less than (to_date('1994-07-01','YYYY-MM-DD'))
    tablespace ts_17,
    partition item32 values less than (to_date('1994-08-01','YYYY-MM-DD'))
    tablespace ts_18,
    partition item33 values less than (to_date('1994-09-01','YYYY-MM-DD'))
    tablespace ts_19,
    partition item34 values less than (to_date('1994-10-01','YYYY-MM-DD'))
    tablespace ts_110,
    partition item35 values less than (to_date('1994-11-01','YYYY-MM-DD'))
    tablespace ts_111,
    partition item36 values less than (to_date('1994-12-01','YYYY-MM-DD'))
    tablespace ts_112,
    partition item37 values less than (to_date('1995-01-01','YYYY-MM-DD'))
    tablespace ts_11,
```

```

partition item38 values less than (to_date('1995-02-01','YYYY-MM-DD'))
tablespace ts_12,
partition item39 values less than (to_date('1995-03-01','YYYY-MM-DD'))
tablespace ts_13,
partition item40 values less than (to_date('1995-04-01','YYYY-MM-DD'))
tablespace ts_14,
partition item41 values less than (to_date('1995-05-01','YYYY-MM-DD'))
tablespace ts_15,
partition item42 values less than (to_date('1995-06-01','YYYY-MM-DD'))
tablespace ts_16,
partition item43 values less than (to_date('1995-07-01','YYYY-MM-DD'))
tablespace ts_17,
partition item44 values less than (to_date('1995-08-01','YYYY-MM-DD'))
tablespace ts_18,
partition item45 values less than (to_date('1995-09-01','YYYY-MM-DD'))
tablespace ts_19,
partition item46 values less than (to_date('1995-10-01','YYYY-MM-DD'))
tablespace ts_110,
partition item47 values less than (to_date('1995-11-01','YYYY-MM-DD'))
tablespace ts_111,
partition item48 values less than (to_date('1995-12-01','YYYY-MM-DD'))
tablespace ts_112,
partition item49 values less than (to_date('1996-01-01','YYYY-MM-DD'))
tablespace ts_11,
partition item50 values less than (to_date('1996-02-01','YYYY-MM-DD'))
tablespace ts_12,
partition item51 values less than (to_date('1996-03-01','YYYY-MM-DD'))
tablespace ts_13,
partition item52 values less than (to_date('1996-04-01','YYYY-MM-DD'))
tablespace ts_14,
partition item53 values less than (to_date('1996-05-01','YYYY-MM-DD'))
tablespace ts_15,
partition item54 values less than (to_date('1996-06-01','YYYY-MM-DD'))
tablespace ts_16,
partition item55 values less than (to_date('1996-07-01','YYYY-MM-DD'))
tablespace ts_17,
partition item56 values less than (to_date('1996-08-01','YYYY-MM-DD'))
tablespace ts_18,
partition item57 values less than (to_date('1996-09-01','YYYY-MM-DD'))
tablespace ts_19,
partition item58 values less than (to_date('1996-10-01','YYYY-MM-DD'))
tablespace ts_110,
partition item59 values less than (to_date('1996-11-01','YYYY-MM-DD'))
tablespace ts_111,
partition item60 values less than (to_date('1996-12-01','YYYY-MM-DD'))
tablespace ts_112,
partition item61 values less than (to_date('1997-01-01','YYYY-MM-DD'))
tablespace ts_11,
partition item62 values less than (to_date('1997-02-01','YYYY-MM-DD'))
tablespace ts_12,
partition item63 values less than (to_date('1997-03-01','YYYY-MM-DD'))
tablespace ts_13,
partition item64 values less than (to_date('1997-04-01','YYYY-MM-DD'))
tablespace ts_14,
partition item65 values less than (to_date('1997-05-01','YYYY-MM-DD'))
tablespace ts_15,
partition item66 values less than (to_date('1997-06-01','YYYY-MM-DD'))
tablespace ts_16,
partition item67 values less than (to_date('1997-07-01','YYYY-MM-DD'))
tablespace ts_17,
partition item68 values less than (to_date('1997-08-01','YYYY-MM-DD'))
tablespace ts_18,
partition item69 values less than (to_date('1997-09-01','YYYY-MM-DD'))
tablespace ts_19,
partition item70 values less than (to_date('1997-10-01','YYYY-MM-DD'))
tablespace ts_110,
partition item71 values less than (to_date('1997-11-01','YYYY-MM-DD'))
tablespace ts_111,
partition item72 values less than (to_date('1997-12-01','YYYY-MM-DD'))
tablespace ts_112,
partition item73 values less than (to_date('1998-01-01','YYYY-MM-DD'))
tablespace ts_11,
partition item74 values less than (to_date('1998-02-01','YYYY-MM-DD'))
tablespace ts_12,
partition item75 values less than (to_date('1998-03-01','YYYY-MM-DD'))
tablespace ts_13,
partition item76 values less than (to_date('1998-04-01','YYYY-MM-DD'))
tablespace ts_14,
partition item77 values less than (to_date('1998-05-01','YYYY-MM-DD'))

```

```

tablespace ts_15,
partition item78 values less than (to_date('1998-06-01','YYYY-MM-DD'))
tablespace ts_16,
partition item79 values less than (to_date('1998-07-01','YYYY-MM-DD'))
tablespace ts_17,
partition item80 values less than (to_date('1998-08-01','YYYY-MM-DD'))
tablespace ts_18,
partition item81 values less than (to_date('1998-09-01','YYYY-MM-DD'))
tablespace ts_19,
partition item82 values less than (to_date('1998-10-01','YYYY-MM-DD'))
tablespace ts_110,
partition item83 values less than (to_date('1998-11-01','YYYY-MM-DD'))
tablespace ts_111,
partition item84 values less than (MAXVALUE)
tablespace ts_112)
as select * from lineitem_et;
drop table orders;
create table orders(
    o_orderdate      ,
    o_orderkey       NOT NULL,
    o_custkey        NOT NULL,
    o_orderpriority  ,
    o_shippriority   ,
    o_clerk          ,
    o_orderstatus    ,
    o_totalprice     ,
    o_comment
)
pctfree 1
pctused 99
initrans 10
compress
parallel(degree 8 instances 4)
nologging
partition by range (o_orderdate)
subpartition by hash(o_custkey)
subpartitions 16
(
    partition ord1 values less than (to_date('1992-01-01','YYYY-MM-DD'))
    tablespace ts_o1,
    partition ord2 values less than (to_date('1992-02-01','YYYY-MM-DD'))
    tablespace ts_o2,
    partition ord3 values less than (to_date('1992-03-01','YYYY-MM-DD'))
    tablespace ts_o3,
    partition ord4 values less than (to_date('1992-04-01','YYYY-MM-DD'))
    tablespace ts_o4,
    partition ord5 values less than (to_date('1992-05-01','YYYY-MM-DD'))
    tablespace ts_o5,
    partition ord6 values less than (to_date('1992-06-01','YYYY-MM-DD'))
    tablespace ts_o6,
    partition ord7 values less than (to_date('1992-07-01','YYYY-MM-DD'))
    tablespace ts_o7,
    partition ord8 values less than (to_date('1992-08-01','YYYY-MM-DD'))
    tablespace ts_o8,
    partition ord9 values less than (to_date('1992-09-01','YYYY-MM-DD'))
    tablespace ts_o9,
    partition ord10 values less than (to_date('1992-10-01','YYYY-MM-DD'))
    tablespace ts_o10,
    partition ord11 values less than (to_date('1992-11-01','YYYY-MM-DD'))
    tablespace ts_o11,
    partition ord12 values less than (to_date('1992-12-01','YYYY-MM-DD'))
    tablespace ts_o12,
    partition ord13 values less than (to_date('1993-01-01','YYYY-MM-DD'))
    tablespace ts_o1,
    partition ord14 values less than (to_date('1993-02-01','YYYY-MM-DD'))
    tablespace ts_o2,
    partition ord15 values less than (to_date('1993-03-01','YYYY-MM-DD'))
    tablespace ts_o3,
    partition ord16 values less than (to_date('1993-04-01','YYYY-MM-DD'))
    tablespace ts_o4,
    partition ord17 values less than (to_date('1993-05-01','YYYY-MM-DD'))
    tablespace ts_o5,
    partition ord18 values less than (to_date('1993-06-01','YYYY-MM-DD'))
    tablespace ts_o6,
    partition ord19 values less than (to_date('1993-07-01','YYYY-MM-DD'))
    tablespace ts_o7,
    partition ord20 values less than (to_date('1993-08-01','YYYY-MM-DD'))
    tablespace ts_o8,
    partition ord21 values less than (to_date('1993-09-01','YYYY-MM-DD'))

```

```

tablespace ts_o9,
partition ord22 values less than (to_date('1993-10-01','YYYY-MM-DD'))
tablespace ts_o10,
partition ord23 values less than (to_date('1993-11-01','YYYY-MM-DD'))
tablespace ts_o11,
partition ord24 values less than (to_date('1993-12-01','YYYY-MM-DD'))
tablespace ts_o12,
partition ord25 values less than (to_date('1994-01-01','YYYY-MM-DD'))
tablespace ts_o1,
partition ord26 values less than (to_date('1994-02-01','YYYY-MM-DD'))
tablespace ts_o2,
partition ord27 values less than (to_date('1994-03-01','YYYY-MM-DD'))
tablespace ts_o3,
partition ord28 values less than (to_date('1994-04-01','YYYY-MM-DD'))
tablespace ts_o4,
partition ord29 values less than (to_date('1994-05-01','YYYY-MM-DD'))
tablespace ts_o5,
partition ord30 values less than (to_date('1994-06-01','YYYY-MM-DD'))
tablespace ts_o6,
partition ord31 values less than (to_date('1994-07-01','YYYY-MM-DD'))
tablespace ts_o7,
partition ord32 values less than (to_date('1994-08-01','YYYY-MM-DD'))
tablespace ts_o8,
partition ord33 values less than (to_date('1994-09-01','YYYY-MM-DD'))
tablespace ts_o9,
partition ord34 values less than (to_date('1994-10-01','YYYY-MM-DD'))
tablespace ts_o10,
partition ord35 values less than (to_date('1994-11-01','YYYY-MM-DD'))
tablespace ts_o11,
partition ord36 values less than (to_date('1994-12-01','YYYY-MM-DD'))
tablespace ts_o12,
partition ord37 values less than (to_date('1995-01-01','YYYY-MM-DD'))
tablespace ts_o1,
partition ord38 values less than (to_date('1995-02-01','YYYY-MM-DD'))
tablespace ts_o2,
partition ord39 values less than (to_date('1995-03-01','YYYY-MM-DD'))
tablespace ts_o3,
partition ord40 values less than (to_date('1995-04-01','YYYY-MM-DD'))
tablespace ts_o4,
partition ord41 values less than (to_date('1995-05-01','YYYY-MM-DD'))
tablespace ts_o5,
partition ord42 values less than (to_date('1995-06-01','YYYY-MM-DD'))
tablespace ts_o6,
partition ord43 values less than (to_date('1995-07-01','YYYY-MM-DD'))
tablespace ts_o7,
partition ord44 values less than (to_date('1995-08-01','YYYY-MM-DD'))
tablespace ts_o8,
partition ord45 values less than (to_date('1995-09-01','YYYY-MM-DD'))
tablespace ts_o9,
partition ord46 values less than (to_date('1995-10-01','YYYY-MM-DD'))
tablespace ts_o10,
partition ord47 values less than (to_date('1995-11-01','YYYY-MM-DD'))
tablespace ts_o11,
partition ord48 values less than (to_date('1995-12-01','YYYY-MM-DD'))
tablespace ts_o12,
partition ord49 values less than (to_date('1996-01-01','YYYY-MM-DD'))
tablespace ts_o1,
partition ord50 values less than (to_date('1996-02-01','YYYY-MM-DD'))
tablespace ts_o2,
partition ord51 values less than (to_date('1996-03-01','YYYY-MM-DD'))
tablespace ts_o3,
partition ord52 values less than (to_date('1996-04-01','YYYY-MM-DD'))
tablespace ts_o4,
partition ord53 values less than (to_date('1996-05-01','YYYY-MM-DD'))
tablespace ts_o5,
partition ord54 values less than (to_date('1996-06-01','YYYY-MM-DD'))
tablespace ts_o6,
partition ord55 values less than (to_date('1996-07-01','YYYY-MM-DD'))
tablespace ts_o7,
partition ord56 values less than (to_date('1996-08-01','YYYY-MM-DD'))
tablespace ts_o8,
partition ord57 values less than (to_date('1996-09-01','YYYY-MM-DD'))
tablespace ts_o9,
partition ord58 values less than (to_date('1996-10-01','YYYY-MM-DD'))
tablespace ts_o10,
partition ord59 values less than (to_date('1996-11-01','YYYY-MM-DD'))
tablespace ts_o11,
partition ord60 values less than (to_date('1996-12-01','YYYY-MM-DD'))
tablespace ts_o12,

```

```

partition ord61 values less than (to_date('1997-01-01','YYYY-MM-DD'))
tablespace ts_o1,
partition ord62 values less than (to_date('1997-02-01','YYYY-MM-DD'))
tablespace ts_o2,
partition ord63 values less than (to_date('1997-03-01','YYYY-MM-DD'))
tablespace ts_o3,
partition ord64 values less than (to_date('1997-04-01','YYYY-MM-DD'))
tablespace ts_o4,
partition ord65 values less than (to_date('1997-05-01','YYYY-MM-DD'))
tablespace ts_o5,
partition ord66 values less than (to_date('1997-06-01','YYYY-MM-DD'))
tablespace ts_o6,
partition ord67 values less than (to_date('1997-07-01','YYYY-MM-DD'))
tablespace ts_o7,
partition ord68 values less than (to_date('1997-08-01','YYYY-MM-DD'))
tablespace ts_o8,
partition ord69 values less than (to_date('1997-09-01','YYYY-MM-DD'))
tablespace ts_o9,
partition ord70 values less than (to_date('1997-10-01','YYYY-MM-DD'))
tablespace ts_o10,
partition ord71 values less than (to_date('1997-11-01','YYYY-MM-DD'))
tablespace ts_o11,
partition ord72 values less than (to_date('1997-12-01','YYYY-MM-DD'))
tablespace ts_o12,
partition ord73 values less than (to_date('1998-01-01','YYYY-MM-DD'))
tablespace ts_o1,
partition ord74 values less than (to_date('1998-02-01','YYYY-MM-DD'))
tablespace ts_o2,
partition ord75 values less than (to_date('1998-03-01','YYYY-MM-DD'))
tablespace ts_o3,
partition ord76 values less than (to_date('1998-04-01','YYYY-MM-DD'))
tablespace ts_o4,
partition ord77 values less than (to_date('1998-05-01','YYYY-MM-DD'))
tablespace ts_o5,
partition ord78 values less than (to_date('1998-06-01','YYYY-MM-DD'))
tablespace ts_o6,
partition ord79 values less than (to_date('1998-07-01','YYYY-MM-DD'))
tablespace ts_o7,
partition ord80 values less than (to_date('1998-08-01','YYYY-MM-DD'))
tablespace ts_o8,
partition ord81 values less than (to_date('1998-09-01','YYYY-MM-DD'))
tablespace ts_o9,
partition ord82 values less than (to_date('1998-10-01','YYYY-MM-DD'))
tablespace ts_o10,
partition ord83 values less than (to_date('1998-11-01','YYYY-MM-DD'))
tablespace ts_o11,
partition ord84 values less than (MAXVALUE)
tablespace ts_o12)
as select * from orders_et;
drop table partsupp;
create table partsupp(
    ps_partkey      NOT NULL,
    ps_supplekey    NOT NULL,
    ps_supplycost    NOT NULL,
    ps_availqty      ,
    ps_comment
)
pctfree 1
pctused 99
tablespace ts_ps
parallel(degree 8 instances 4)
nologging
partition by hash (ps_supplekey)
partitions 16
as select * from partsupp_et
;
drop table part;
create table part(
    p_partkey      NOT NULL,
    p_type         ,
    p_size         ,
    p_brand        ,
    p_name         ,
    p_container    ,
    p_mfgr         ,
    p_retailprice   ,
    p_comment
)
pctfree 1

```

```

pctused 99
tablespace ts_p
parallel(degree 8 instances 4)
nologging
partition by hash (p_partkey)
partitions 16
as select * from part_et
;
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 1
pctused 99
tablespace ts_c
parallel(degree 8 instances 4)
nologging
partition by hash (c_custkey)
partitions 16
as select * from customer_et
;
drop table supplier;
create table supplier(
  s_suppkey      NOT NULL,
  s_nationkey    ,
  s_comment      ,
  s_name         ,
  s_address      ,
  s_phone        ,
  s_acctbal
)
pctfree 1
pctused 99
tablespace ts_s
parallel(degree 8 instances 4)
nologging
partition by hash (s_suppkey)
partitions 16
as select * from supplier_et
;
drop table nation;
create table nation(
  n_nationkey    NOT NULL,
  n_name         ,
  n_regionkey    ,
  n_comment
)
tablespace ts_s
parallel(degree 1 instances 4)
as select * from nation_et
;
drop table region;
create table region(
  r_regionkey    ,
  r_name         ,
  r_comment
)
tablespace ts_s
parallel(degree 1 instances 4)
as select * from region_et
;
drop table lineitem;
drop table orders;
drop table part;
drop table supplier;
drop table partsupp;
drop table customer;
drop table nation;
drop table region;
spool off;
exit;
EOF

```

create_indexes_all.sh

```

#!/bin/ksh

# Create All TPC-H Schema Indexes...

sqlplus tpcd/tpcd <<EOF
spool create_indexes_all_spool.log;
drop index i_ps_partkey_supkey;
create unique index i_ps_partkey_supkey
on partsupp (ps_partkey,ps_supkey)
pctfree 2
initrans 10
nologging
compute statistics
tablespace ts_i_ps
parallel (degree 8 instances 4);
drop index i_c_custkey;
create unique index i_c_custkey
on customer (c_custkey)
pctfree 2
initrans 10
nologging
compute statistics
tablespace ts_i_c
parallel (degree 8 instances 4);
drop index i_o_orderkey;
create unique index i_o_orderkey
on orders (o_orderkey)
pctfree 2
initrans 10
nologging
compute statistics
tablespace ts_i_o
parallel (degree 8 instances 4);
drop index i_l_orderkey;
create index i_l_orderkey
on lineitem (l_orderkey)
pctfree 2
initrans 10
nologging
compute statistics
tablespace ts_i_l
parallel (degree 8 instances 4);
spool off;
EOF

```

analyze_ops.sh

```

#!/bin/ksh

echo "Starting Database Analyze Step..." `date`

sqlplus tpcd/tpcd <<EOF
spool analyze_ops_spool.log;
rem alter session set parallel_instance_group = 'groupb';
execute dbms_stats.gather_schema_stats('tpcd', estimate_percent => 1, degree => 64
, granularity => 'GLOBAL')
spool off;
EOF

echo "Completed Database Analyze Step..." `date`

```

ACID Test Source Code

atom.sh

```

#!/bin/ksh
#
# $Header: atom.sh 05-sep-2001.08:03:33 mpoess Exp $
#
# atom.sh

```

```

#
# Copyright (c) 1999, 2001, Oracle Corporation. 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    09/05/01 - reduce lines
#   mpoess    07/10/01 - change scale factor to 1
#   mpoess    08/08/99 - Creation
#   mpoess    08/08/99 - Creation
#

. $KIT_DIR/env

ITER=3
SF=1
PROG=atranspl
OUT=${ACID_OUT}/atom
USER=${DATABASE_USER}

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

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

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

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

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

```

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_linenum = 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_newprice := l_eprice + cost;
        o_newtprice := trunc((l_newprice * (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_newprice,
                l_quantity = l_newquan
            where l_orderkey = o_key
            and l_linenum = l_key;

        update orders
            set o_totalprice = o_newtprice
            where o_orderkey = o_key;
    END LOOP;
END;

```

```

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;

```

atranspl.c

/* Copyright (c) 2001, 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 10/17/01 - 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 gettimeofday();
void sql_error();
void usage();
void ACIDinit();
void ACIDexit();
int atoi();
void srand48();
long lrand48();

```

/* declarations for ORDERS */

```

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

```

/* declarations for LINEITEM */

```

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

```

```

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

```

```

sb2 l_nprice;

```

/* 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>] [o<pathname for output>] [d<pathname for
durability file>] [u<uid/passwd>] \n\n");

    fprintf(stderr, "  proc_no    :the process number within this ACID\n");
    fprintf(stderr, "  num_streams :the total number of ACID transaction streams\n");
    fprintf(stderr, "  commit      :1 to commit transaction, abort otherwise\n");
    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);
    OCIlfree(tpcenv, OCI_HTYPE_STMT);
    OCIlfree(tpcsvc, OCI_HTYPE_SVCCTX);
    OCIlfree(tpcsrv, OCI_HTYPE_SERVER);
    OCIlfree(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) OCIErrGet(errhp, 1, NULL, (sb4 *) &errcode, (text *) msg,
                2048, OCI_HTYPE_ERROR);
        else
            (void) OCIErrGet(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) OCIErrGet(errhp, 1, NULL, (sb4 *) &errcode, (text *) msg,
                2048, OCI_HTYPE_ERROR);
        else
            (void) OCIErrGet(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) OCIErrGet(errhp, 1, NULL, (sb4 *) &errcode, (text *) msg,
                2048, OCI_HTYPE_ERROR);
        else
            (void) OCIErrGet(errhp, 1, NULL, (sb4 *) &errcode, (text *) msg,
                2048, OCI_HTYPE_ENV);
        fprintf(stderr, "%s\n", msg);
        break;
    }
    /* Rollback just in case */

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

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

    ACIDexit();

    exit(1);
}

#ifdef LINUX
int main(argc, argv)
#else
void main(argc, argv)
#endif
    int argc;
    char *argv[];
{
    int i;
    char line[64];
    ub4 errcode;
    char msg[2048];
    int need_commit = 0;

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

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

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

    proc_no = atoi(argv[1]);

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

    num_streams = atoi(argv[2]);

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

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

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

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

    /* Process optional parameters */

    argc -= 4;
    argv += 4;

```



```

while(--argc) {
    ++argv;
    switch(argv[0][0]) {
        case 'u':
            strncpy((char *) lname, ++(argv[0]), UNAME_LEN);
            if (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
    sscanf(line, "%d %d\n", &o_key, &delta);

    /* Obtain l_key from l_key query */

    OCIsexec(tpcscv, curi, errhp, 1);

```

```

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

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

/* Generate delta if necessary */

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

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

curr_time = time(NULL);

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

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

OCIsexec(tpcscv, curr, errhp, 1);

curr_time = time(NULL);

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

    OCIsExec(tpcscv, cure1, errhp, 1);
    OCIsExec(tpcscv, cure2, errhp, 1);

    FPRTF1(outfile, "AFTER TRANSACTION:\n");
    FPRTF1(outfile, "l_extendedprice: %.2f\n", l_newprice);
    FPRTF1(outfile, "l_quantity:    %d\n", (int) l_newquan);
    FPRTF1(outfile, "o_totalprice:  %.2f\n\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);
    FPRTF1(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 = OCIEEnvInit((OCIEnv **) &tpcenv, OCI_DEFAULT, 0, (dvoid **)0)) !=
        OCI_SUCCESS)
        sql_error(tpcenv, status, 0);

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

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

```

```

    exit(-1);
}
*/

/* get username and password */

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

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

OCIAsSet(tpcscv, OCI_HTYPE_SVCCTX, tpcsrv, 0, OCI_ATTR_SERVER, errhp);

OCIAsSet(tpcusr, OCI_HTYPE_SESSION, lname, strlen(lname), OCI_ATTR_USERNA
ME,
    errhp);

OCIAsSet(tpcusr, OCI_HTYPE_SESSION, passwd, strlen(passwd), OCI_ATTR_PASSW
ORD,
    errhp);

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

OCIAsSet(tpcscv, OCI_HTYPE_SVCCTX, tpcusr, 0, OCI_ATTR_SESSION, errhp);

/* Enable session parallel dml */

sprintf((char *) sqlstmt, PDMLTXT);
OCISmtPrepare(curi, errhp, (text *)sqlstmt, strlen((char *)sqlstmt),
    OCI_NTV_SYNTAX, OCI_DEFAULT);
OCIsExec(tpcscv, curi, errhp, 1);

/* Enable session parallel ddl */

/*sprintf((char *) sqlstmt, PDDLTX);
OCISmtPrepare(curi, errhp, (text *)sqlstmt, strlen((char *)sqlstmt),
    OCI_NTV_SYNTAX, OCI_DEFAULT);
OCIsExec(tpcscv, curi, errhp, 1);*/

/* Make session serializable */

sprintf((char *) sqlstmt, ISOTXT);
OCISmtPrepare(curi, errhp, (text *)sqlstmt, strlen((char *)sqlstmt),
    OCI_NTV_SYNTAX, OCI_DEFAULT);
OCIsExec(tpcscv, curi, errhp, 1);

/* Set optimizer_index_cost_adj = 25 */

sprintf((char *) sqlstmt, OICATXT);
OCISmtPrepare(curi, errhp, (text *)sqlstmt, strlen((char *)sqlstmt),
    OCI_NTV_SYNTAX, OCI_DEFAULT);
OCIsExec(tpcscv, curi, errhp, 1);

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

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

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

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

#endif /* NOLKEY */

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

```

```

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

/* bind variables */

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

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

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

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

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

/* bind variables */

OCIBbname(cure1,l_newewprice1_bp,errhp,":l_newewprice",ADR(l_newewprice),
    SIZ(l_newewprice),SQLT_FLT);
OCIBbname(cure1,l_newquan1_bp,errhp,":l_newquan",ADR(l_newquan),
    SIZ(l_newquan),SQLT_INT);
OCIBbname(cure1,o_key1_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),SQLT_INT);
OCIBbname(cure1,l_key1_bp,errhp,":l_key",ADR(l_key),SIZ(l_key),SQLT_INT);

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

atranspl.h

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

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

DESCRIPTION

MODIFIED (MM/DD/YY)
mpoess 10/17/01 - add TXT parameter
mpoess 04/09/01 - add hint to find max linenumbr
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>
#ifndef OCIDFN
#include <ocidfn.h>
#endif /* OCIDFN */

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

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

extern int errno;

#ifndef NULL
#define NULL 0
#endif

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

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#ifndef ub1
#define ub1 unsigned char
#endif

#define UNAME_LEN 64
#define WRITE_BUF_LEN 1024

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

#define ADR(object) ((ub1 *)&(object))
#define SIZ(object) ((sword)sizeof(object))
#define BIS(flag,mask) (unsigned) (flag | (unsigned) mask)
#define BIT(flag,mask) (unsigned) ((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 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 OCIsset(hndl,htyp,attp,size,atyp,errh) \
    if((status=OCIAttrSet((dvoid *)hndl,htyp,(dvoid *)attp,size,atyp,errh)) != \
    OCI_SUCCESS) \
        sql_error(errh,status,1); \
    else \
        DISCARD 0

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

#define OCIsbname(stmh,bindp,errh,sqlvar,progv,progv1,ftype) \
    if((status=OCIBindByName(stmh,&bindp,errh,(text *)sqlvar,strlen(sqlvar), \
        progv,progv1,ftype,0,0,0,0,OCI_DEFAULT)) != OCI_SUCCESS) \
        sql_error(errh,status,1); \
    else \
        DISCARD 0

#define OCIsbnamei(stmh,bindp,errh,sqlvar,progv,progv1,ftype,indp) \
    if((status=OCIHandleAlloc((dvoid *)stmh,(dvoid **)&bindp,OCI_HTYPE_BIND, \
        0,(dvoid **)0))!=OCI_SUCCESS) \
        sql_error(stmh,status,0); \
    if((status=OCIBindByName(stmh,&bindp,errh,(text *)sqlvar,strlen(sqlvar), \
        progv,progv1,ftype,indp,0,0,0,0,OCI_DEFAULT)) != OCI_SUCCESS) \
        sql_error(errh,status,1); \
    else \
        DISCARD 0

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

#define OCIsrol(svcv,errh) \
    if((status=OCITransRollback(svcv,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 4)"
#define PDDLTX "alter session force parallel ddl parallel (degree 4)"
#define OICATXT "alter session set optimizer_index_cost_adj=25"

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

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

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

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

```

```
#endif /* ATRANSPL_H */
```

ckpt.sh

```

#!/bin/ksh
#
# $Header: ckpt.sh 10-jul-2001.11:12:03 mpoess Exp $
#
# ckpt.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation. 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 07/10/01 - change svrmgrl to sqlplus
#   mpoess 08/08/99 - Creation
#   mpoess 08/08/99 - Creation
#

```

```
. $KIT_DIR/env
```

```
sqlplus -s /NOLOG<< !
```

```

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

```

```
!
```

cnt_hist.sql

```

select count(*) from history;
exit;

```

consist.sh

```

#!/bin/ksh
#
# $Header: consist.sh 05-sep-2001.08:09:57 mpoess Exp $
#
# consist.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation. 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 09/05/01 - reduce lines
#   mpoess 07/10/01 - add automatic stream setting
#   mpoess 08/08/99 - Creation
#   mpoess 08/08/99 - Creation
#

```

```

.$KIT_DIR/env

KEY=${ACID_DIR}/key$$
OUTFILE=${ACID_OUT}/consrte
CON1=${ACID_OUT}/conb
CON2=${ACID_OUT}/cona
CHK=${ACID_OUT}/consckpt
SF=1

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

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

STREAM=5
let STREAM="$STREAM + 1" # add one for the update stream
ITER=100
PROG=atranspl
USER=${DATABASE_USER}
CK=10

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

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

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

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

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

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

i=0

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

```

```

done

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

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

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

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

i=0

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

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

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

wait

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

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

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

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

i=0
while [ $i -lt $STREAM ]
do
    KEYS=`head -10 ${KEY}$i | awk '{printf "%d ", $1}`
    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: template.sql 16-feb-2001.19:27:01 mpoess  Exp $
Rem
Rem consist.sql
Rem
Rem Copyright (c) Oracle Corporation 2001. 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 NOTES
Rem REQUIRES PACKAGES prvtotpt and dbmsotpt
Rem
Rem MODIFIED (MM/DD/YY)
Rem mpoess 07/10/01 - Created
Rem

SET FEEDBACK 1
SET NUMWIDTH 10
SET LINESIZE 80
SET TRIMSPOOL ON
SET TAB OFF
SET PAGESIZE 100
SET ECHO ON

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

```

count_tx.sh

```

#!/bin/ksh

STEM=$1
ITER=$2
OUT=$3
FIN=FALSE
while [ "$FIN" = "FALSE" ]
do

```

```

s=0
FIN=TRUE
while [ $s -lt $STEM ]
do
    nt=`grep "Transaction Completed" $OUT/dura${s} | wc -l`
    if [ $nt -lt $ITER ];then
        FIN=FALSE
    fi
    s=`expr $s + 1`
done
sleep 5
done
echo all streams have committed $ITER transactions

```

d_hist.sql

```

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

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

drop table history;

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

exit;

```

end_acid.sh

```

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

```

```

# MODIFIED (MM/DD/YY)
# mpoess 08/08/99 - Creation
# mpoess 08/08/99 - Creation
#

. $KIT_DIR/env

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

ITER=100
STEM=6
PROG=${ACID_DIR}/atranspl.ott
IN=${ACID_DIR}/acid_in
DURA=${DURA_DIR}/dura
OUT=${DURA_DIR}/drate
DSMPL=${DURA_DIR}/durasmpl
KEY=${DURA_DIR}/key${1}_
USER=tpcd/tpcd
TRIG=1
HCNT=duracnta

# get history count

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

# perform the consistency

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

gettime.c

#ifdef RCSID
static char *RCSid =
    "$Header: gettime.c 15-jul-99.14:27:44 mpoess Exp $ ";
#endif /* RCSID */

/* Copyright (c) Oracle Corporation 1999. All Rights Reserved. */

/*

NAME
    gettime.c

DESCRIPTION
    get wall clock time.
    get cpu time.

FUNCTIONS
    get wall clock time.
    get cpu time.

NOTES
    Both routines return time in seconds as a double.
MODIFIED (MM/DD/YY)
mpoess 07/15/99 - Creation
mpoess 07/15/99 - Creation
espeed 01/15/02 - Customize for Compaq Tru64 UNIX

*/

```

```

/*
** Options:
** TIME_W_TIMES:    implement gettimeofday() with times().
** TIME_W_GETTIME:  implement gettimeofday() with gettimeofday().
** CPU_W_TIMES:     implement getcpu() with times().
** CPU_W_GETTRU:    implement getcpu() with getrusage().
** GETRU_STATS:     collect getrusage statistics
** GET_P_STATS:     collect get_process_stats statistics
*/

#define SUN_OS5

#ifdef SUN_OS5
#define TIME_W_GETTIME
#define CPU_W_TIMES
#undef GETRU_STATS
#undef CPU_W_GETTRU
#endif /* SUN_OS5 */

#ifdef sequent || defined(SEQ_PXS)
#define GET_P_STATS
#endif /* sequent */

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

#ifdef a_osf || defined(A_OSF)
#define TIME_W_GETTIME
#define CPU_W_GETTRU
#define GETRU_STATS
#endif /* AIXRIOS */

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

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

#ifdef !defined(CPU_W_GETTRU) && !defined(CPU_W_TIMES)
#define CPU_W_TIMES
#endif

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

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

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

#ifdef TIME_W_TIMES || defined(CPU_W_TIMES)
#include <sys/types.h>
#include <sys/times.h>
/* # include <sys/param.h> most systems define HZ here */
/* but Compaq Tru64 UNIX requires the following 2 includes */
#include <sys/sysinfo.h>
#include <machine/hal_sysinfo.h>
#endif /* TIME_W_TIMES or CPU_W_TIMES */

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

```

```

#include <stdio.h>

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

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

double gettime ()
{
#ifdef TIME_W_GETTIME
    struct timeval tv;

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

#ifdef TIME_W_TIMES
    struct tms buf;

    /* Next 2 lines for Compaq Tru64 UNIX */
    int HZ;
    getsysinfo(GSI_CLK_TCK, (char *)&HZ, sizeof(HZ));

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

}

double getcpu ()
{
#ifdef CPU_W_TIMES
    struct tms buf;

    /* Next 2 lines for Compaq Tru64 UNIX */
    int HZ;
    getsysinfo(GSI_CLK_TCK, (char *)&HZ, sizeof(HZ));

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

#ifdef CPU_W_GETRU
    struct rusage ru;
    double usecs;

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

}

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

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

{
#ifdef GETRU_STATS

```

```

    struct rusage ru;

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

#ifdef GET_P_STATS
    timeval_t tv;
    struct process_stats ru;

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

}

getru1 (kids)

int kids;

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

#ifdef GET_P_STATS
    timeval_t tv;

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

}

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

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

{
#ifdef GETRU_STATS
    struct rusage ru;

    fprintf (fp, "%-10.10s %-10.10s %10d %10d ", config, runname, proc_no, kids);
    getrusage (kids ? RUSAGE_CHILDREN : RUSAGE_SELF, &ru);
    if (kids)
        diffrru (&ru, &kidsru);
    else
        diffrru (&ru, &selfru);
    print_ru (fp, &ru);

```



```

    fprintf (fp, "\n");
#endif /* GETRU_STATS */

#ifdef GET_P_STATS
    timeval_t tv;
    struct process_stats ru;

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

}

#ifdef GETRU_STATS

print_ru (fp, ru)

FILE *fp;
struct rusage *ru;

{

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

}

diffpu (ru2, ru)

struct rusage *ru2;
struct rusage *ru;

{

    ru2->ru_ftime.tv_sec -= ru->ru_ftime.tv_sec;
    ru2->ru_ftime.tv_usec -= ru->ru_ftime.tv_usec;
    ru2->ru_stime.tv_sec -= ru->ru_stime.tv_sec;
    ru2->ru_stime.tv_usec -= ru->ru_stime.tv_usec;
    ru2->ru_maxrss -= ru->ru_maxrss;
    ru2->ru_ixrss -= ru->ru_ixrss;
    ru2->ru_idrss -= ru->ru_idrss;
    ru2->ru_minflt -= ru->ru_minflt;
    ru2->ru_majflt -= ru->ru_majflt;
    ru2->ru_nswap -= ru->ru_nswap;
    ru2->ru_inblock -= ru->ru_inblock;
    ru2->ru_oublock -= ru->ru_oublock;
    ru2->ru_msgsnd -= ru->ru_msgsnd;

    ru2->ru_msgrcv -= ru->ru_msgrcv;
    ru2->ru_nsignals -= ru->ru_nsignals;
    ru2->ru_nvcsw -= ru->ru_nvcsw;
    ru2->ru_nivcsw -= ru->ru_nivcsw;

}

#endif /* GETRU_STATS */

#ifdef GET_P_STATS

print_ru (fp, ps)

FILE *fp;
struct process_stats *ps;

{

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

}

diffpu (ru2, ru)

struct process_stats *ru2;
struct process_stats *ru;

{

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

}

#endif /* GET_P_STATS */

```

iso1.sh

```
#!/bin/ksh
#
# $Header: iso1.sh 10-jul-2001.11:17:34 mpoess Exp $
#
# iso1.sh
#
# Copyright (c) 1998, 2001, Oracle Corporation. 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 07/10/01 - change tpcd/tpcd to DATABASE_USER
#   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=rsh

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; RHOST="$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 "" >> $TXN2FILE
echo "" >> $TXN2FILE
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >> $TXN2FILE
echo "" >> $TXN2FILE
echo "-----" >> $TXN2FILE
```

```
sleep 1
```

```
# start ACID transaction, Sleep for 60 second before COMMIT
```

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

```
# let's sleep 10 seconds before starting ACID query
```

```
sleep 10
```

```
# start ACID query with the same OKEY
```

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

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

```
cat $TXN1FILE $TXN2FILE >> $ISOFILE
```

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

iso2.sh

```
#!/bin/ksh
#
# $Header: iso2.sh 10-jul-2001.11:17:36 mpoess Exp $
#
# iso2.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation. 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 07/10/01 - change tpcd/tpcd to DATABASE_USER
#   mpoess 08/04/99 - Creation
#   mpoess 08/04/99 - Creation
#
#
```

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

. $KIT_DIR/env

RSH=rsh

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

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

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

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

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

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

while :
do
    case "$1" in
        -u) shift; USER=$1;;
        -n) shift; RHOST="$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 "" >> $TXN2FILE
echo "" >> $TXN2FILE
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >> $TXN2FILE
echo "" >> $TXN2FILE
echo "-----" >> $TXN2FILE

sleep 1

# start ACID transaction, Sleep for 30 second before ROLLBACK

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

# let's sleep 10 seconds before starting ACID query

sleep 10

# start ACID query with the same OKEY

echo "Running ACID query 10 seconds AFTER the start of ACID transaction" \
```

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

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

iso3.sh

```
#!/bin/ksh
#
# $Header: iso3.sh 10-jul-2001.11:17:37 mpoess Exp $
#
# iso3.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation. 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 07/10/01 - change tpcd/tpcd to DATABASE_USER
# mpoess 08/04/99 - Creation
# mpoess 08/04/99 - Creation
#
```

```
. $KIT_DIR/env

# May need to change the following:
RSH=rsh

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

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {
    echo ""
    echo "Usage: $0 [-u user/passwd] [-n remote_node] -h"
    echo ""
```

```

    exit 1;
}

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

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

# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

sleep 1

# start ACID transaction, Sleep for 30 second before COMMIT

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

# let's sleep 10 seconds before starting second ACID transaction

sleep 10

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

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

```

iso4.sh

```

#!/bin/ksh
#
# $Header: iso4.sh 10-jul-2001.11:17:38 mpoess Exp $
#
# iso4.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation. 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 07/10/01 - change tpcd/tpcd to DATABASE_USER
# mpoess 08/04/99 - Creation
# mpoess 08/04/99 - Creation
#

```

```
. $KIT_DIR/env
```

```

# May need to change the following:
RSH=rsh

```

```

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

```

```
DURA_DIR=$ACID_DIR/dura
```

```

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

```

```

USER=$DATABASE_USER
PROG=atranspl

```

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

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

```

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

```

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

```

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

```

```
# generate key files
```

```
randkey 1 0.1 u"$USER" > $KEYFILE
```

```
sleep 1
```

```
# start ACID transaction, Sleep for 30 second before ROLLBACK
```

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

```
# let's sleep 10 seconds before starting second ACID transaction
```

```
sleep 10
```

```

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

```

```

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

```

```

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

iso5.sh

#!/bin/ksh
#
# $Header: iso5.sh 10-jul-2001.11:17:39 mpoess Exp $
#
# iso5.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation. 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 07/10/01 - change tpcd/tpcd to DATABASE_USER
#   mpoess 08/04/99 - Creation
#   mpoess 08/04/99 - Creation
#

. $KIT_DIR/env

# May need to change the following:
RSH=rsh

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

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

USER=$DATABASE_USER
PROG=atranspl

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

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

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

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

while :
do
    case "$1" in
        -u) shift; USER=$1;;
        -n) shift; RHOST="$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 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 0.1`

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

wait

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

```

iso6.sh

```

#!/bin/ksh
#
# $Header: iso6.sh 10-jul-2001.11:17:41 mpoess Exp $
#
# iso6.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation. 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    07/10/01 - change tpcd/tpcd to DATABASE_USER
#   mpoess    08/04/99 - Creation
#   mpoess    08/04/99 - Creation
#

. $KIT_DIR/env

# May need to change the following:
RSH=rsh

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

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

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

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

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

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

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

# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

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

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

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

sleep 1

# start Query 17, 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 [ "$RHOST" != "" ]
then
echo "Starting ACID transaction on node $RHOST" >> $TXN2FILE
${RSH} -n ${RHOST} $PROG 1 1 1 0 i$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) Oracle Corporation 2001. All Rights Reserved. */

/*

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

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

*/

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

#define ORDERCNT 150000.0

/* MK_SPARSE adopted from dss.h */

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

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

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

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

long l_key = 0;

```

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

/* OCI handles */

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

OCIBind *l_key_bp;
OCIBind *o_key_bp;

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

char sqlstmt[1024];

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

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

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

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

/* Rollback just in case */

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

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

```

```

fflush(stderr);

ACIDexit();

exit(1);
}

main(argc, argv)
    int argc;
    char **argv;
{
    long count;
    long i;
    double sf; /* need to accomodate sf 0.1 */
    double random;
    double ordcnt;
    adef *res;

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

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

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

    argc -= 2;
    argv += 2;

    while (--argc) {
        ++argv;
        switch(argv[0][0]) {
        case 'u':
            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 drand48() output is 'unique', first get a number within */
        /* the range of [0..sf*ORDERCNT] and then maps the different */
        /* ranges to generate the real output. */

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

        /* Obtain l_key from l_key query */

        OCIsexec(tpcsvc, curi, errhp, 1);

        /* l_key is the highest l_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=OCIEEnvInit((OCIEEnv **)&tpcenv,OCI_DEFAULT,0,(dvoid **)0)) !=
        OCI_SUCCESS)
        sql_error(tpcenv, status, 0);

    OCIAlloc(tpcenv,&errhp,OCI_HTYPE_ERROR);
    OCIAlloc(tpcenv,&curi,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(lname, '/');
    *passwd = '\0';
    passwd++;

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

    OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcsrv,0,OCI_ATTR_SERVER,errhp);

    OCIaset(tpcusr,OCI_HTYPE_SESSION,lname,strlen(lname),OCI_ATTR_USERNA
ME,
        errhp);

    OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI_ATTR_PASSW
ORD,
        errhp);

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

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

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

    sprintf((char *) sqlstmt,SQLTXT1);
    OCISmtPrepare(cur,errhp,(text *)sqlstmt,strlen((char *)sqlstmt),
        OCI_NTV_SYNTAX,OCI_DEFAULT);

    OCIbname(cur,l_key_bp,errhp,":l_key",ADR(l_key),SIZ(l_key),SQLT_INT);
    OCIbname(cur,o_key_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),SQLT_INT);
}

```

randpsup.c

```

/* Copyright (c) Oracle Corporation 2001. 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 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
#

OH=$ORACLE_HOME
OUT_DIR=$ACID_OUT

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

ITER=400
STEM=6
SF=1
PROG=atranspl
IN=${ACID_DIR}/acid_in
DURA_DIR=$ACID_OUT/dura
OUT=$DURA_DIR/drate
DURA=$DURA_DIR/dura
KEY=${DURA_DIR}/key$_
USER=tpcd/tpcd
TRIG=1
HCNT=duracntb

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

# 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;;
    esac
done

--) 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 tpcd/tpcd @consist $j >> $DURA_DIR/duraconsb
    done
    i=`expr $i + 1`
done

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

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

wait

echo "ACID run completed"
```

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

# $1 durability output file

.$KIT_DIR/env
```

sample.sh

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

# $1 durability output file

.$KIT_DIR/env
```

```

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

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

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

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

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

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

```

sample.sql

```

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;

```

Disk Configuration Data

HSG80_storage_array_configuration

HSG80 Configuration/Layout for ORACLE 100GB TPC-H Project

For each of the (12) HSG80 Controller Pair Arrays:

RUN CONFIG

```

ADD MIRRORSET M1 DISK10000 DISK20100
ADD MIRRORSET M2 DISK10100 DISK20000
ADD MIRRORSET M3 DISK30000 DISK40100
ADD MIRRORSET M4 DISK30100 DISK40000
ADD MIRRORSET M5 DISK50000 DISK60100
ADD MIRRORSET M6 DISK50100 DISK60000
ADD MIRRORSET M7 DISK10200 DISK20300
ADD MIRRORSET M8 DISK10300 DISK20200
ADD MIRRORSET M9 DISK30200 DISK40300

```

```

ADD MIRRORSET M10 DISK30300 DISK40200
ADD MIRRORSET M11 DISK50200 DISK60300
ADD MIRRORSET M12 DISK50300 DISK60200

```

```

ADD STRIPESET S1 M1 M2 M3 M4 M5 M6
ADD STRIPESET S2 M7 M8 M9 M10 M11 M12

```

```

INIT S1 (Default CHUNKSIZE = 256 blocks or 128K)
INIT S2

```

```

ADD UNIT D1 S1 WRITEBACK_CACHE
ADD UNIT D101 S2 WRITEBACK_CACHE

```

Note: Each HSG80 UNIT is recognized as a "disk" LUN by the Tru64 UNIX Operating System upon boot.

Logical Storage Manager Scripts

lsm_create.sh

```

./volsetup_disks.sh
./sd_flatfiles.sh
./plex_flatfiles.sh
./sd_orders.sh
./plex_orders.sh
./sd_lineitem_1.sh
./plex_lineitem_1.sh
./sd_files.sh
./plex_files.sh
./sd_temps.sh
./plex_temps.sh
./sd_lineitem_2.sh
./plex_lineitem_2.sh
./sd_system4.sh
./plex_system4.sh
date

```

volsetup_disks.sh

```

#!/bin/sh
set -x
date
#
volsetup -c -o force privlen=32768 dsk2 dsk3 dsk4 dsk5 dsk6 dsk7 dsk8 dsk9 dsk10
dsk11 dsk12 dsk13 dsk19 dsk20 dsk21 dsk22 dsk23 dsk24 dsk25 dsk26 dsk27 dsk28
dsk29 dsk30
date

```

sd_files.sh

```

#!/bin/ksh
set -x
#-----
# sd_files.sh
#-----
echo "start of sd_files"
DDISKS="2 3 4 5 6 7 8 9 10 11 12 13 19 20 21 22 23 24 25 26 27 28 29 30"

SD_NUM=54
SD_SIZE=512000

for DISK in $DDISKS
do
CNT=1
OFFSET=56934400

while [ $CNT -le $SD_NUM ]
do
volmake sd sd${DISK}_files${CNT} dsk${DISK} offset=${OFFSET}
len=${SD_SIZE}
CNT=`expr $CNT + 1`
OFFSET=`expr $OFFSET + $SD_SIZE`
done
done

```

```
echo "end of sd_files.sh"
```

sd_flatfiles.sh

```
#!/bin/ksh
set -x
#-----
# sd_flatfiles.sh
#-----
echo "start of sd_flat"
DDISKS="2 3 4 5 6 7 8 9 10 11 12 13 19 20 21 22 23 24 25 26 27 28 29 30"
SD_NUM=1
SD_SIZE=22528000

for DISK in $DDISKS
do
    CNT=1
    OFFSET=0

    while [ $CNT -le $SD_NUM ]
    do
        volmake sd sd${DISK}_flat${CNT} dsk${DISK} offset=${OFFSET}
len=${SD_SIZE}
        CNT=`expr $CNT + 1`
        OFFSET=`expr $OFFSET + $SD_SIZE`
    done
done

echo "end of sd_flat.sh"
```

sd_lineitem.sh

```
#!/bin/ksh
set -x
#-----
# sd_lineitem.sh
#-----
echo "start of sd_lineitem"
DDISKS="2 3 4 5 6 7 8 9 10 11 12 13 19 20 21 22 23 24 25 26 27 28 29 30"

SD_NUM=24
SD_SIZE=1126400

for DISK in $DDISKS
do
    CNT=1
    OFFSET=29900800

    while [ $CNT -le $SD_NUM ]
    do
        volmake sd sd${DISK}_line${CNT} dsk${DISK} offset=${OFFSET}
len=${SD_SIZE}
        CNT=`expr $CNT + 1`
        OFFSET=`expr $OFFSET + $SD_SIZE`
    done
done

echo "end of sd_lineitem.sh"
```

sd_orders.sh

```
#!/bin/ksh
set -x
#-----
# sd_orders.sh
#-----
echo "start of sd_orders"
DDISKS="2 3 4 5 6 7 8 9 10 11 12 13 19 20 21 22 23 24 25 26 27 28 29 30"
```

```
SD_NUM=24
SD_SIZE=307200
```

```
for DISK in $DDISKS
do
    CNT=1
    OFFSET=22528000

    while [ $CNT -le $SD_NUM ]
    do
        volmake sd sd${DISK}_ord${CNT} dsk${DISK} offset=${OFFSET}
len=${SD_SIZE}
        CNT=`expr $CNT + 1`
        OFFSET=`expr $OFFSET + $SD_SIZE`
    done
done

echo "end of sd_orders.sh"
```

sd_temps.sh

```
#!/bin/ksh
set -x
#-----
# sd_temps.sh
#-----
echo "start of sd_temps"
DDISKS="2 3 4 5 6 7 8 9 10 11 12 13 19 20 21 22 23 24 25 26 27 28 29 30"
SD_NUM=36
SD_SIZE=921600

for DISK in $DDISKS
do
    CNT=1
    OFFSET=84582400

    while [ $CNT -le $SD_NUM ]
    do
        volmake sd sd${DISK}_temps${CNT} dsk${DISK} offset=${OFFSET}
len=${SD_SIZE}
        CNT=`expr $CNT + 1`
        OFFSET=`expr $OFFSET + $SD_SIZE`
    done
done

echo "end of sd_temps.sh"
```

plex_files.sh

```
#!/bin/sh
#
set -x
#
# WIDTH=256
#-----
# plex_files.sh
# Create plexes & vols for regular data files of the same size
#-----
NUM=50
sdNUM=1
while [ $sdNUM -le 54 ]
do

    volmake plex pl_$sdNUM
sd=sd2_files${sdNUM},sd3_files${sdNUM},sd4_files${sdNUM},sd5_files${sdNUM},sd6_files${sdNUM},sd7_files${sdNUM},sd8_files${sdNUM},sd9_files${sdNUM},sd10_files${sdNUM},sd11_files${sdNUM},sd12_files${sdNUM},sd13_files${sdNUM},sd14_files${sdNUM},sd15_files${sdNUM},sd16_files${sdNUM},sd17_files${sdNUM},sd18_files${sdNUM},sd19_files${sdNUM},sd20_files${sdNUM},sd21_files${sdNUM},sd22_files${sdNUM},sd23_files${sdNUM},sd24_files${sdNUM},sd25_files${sdNUM},sd26_files${sdNUM},sd27_files${sdNUM},sd28_files${sdNUM},sd29_files${sdNUM},sd30_files${sdNUM} layout=STRIPE stwidth=$WIDTH ncol=24

    sdNUM=`expr $sdNUM + 1`
done
```

```

volmake -U gen vol vol_$NUM plex=pl_$NUM read_pol=SELECT user=oracle
group=dba mode=0777 log_type=NONE len=0

volume start vol_$NUM
volume init active vol_$NUM
NUM=`expr $NUM + 1`
sdNUM=`expr $sdNUM + 1`
done

echo "end of plex_files.sh"

```

plex_flatfiles.sh

```

#!/bin/sh
#
set -x
#
#
WIDTH=128
#
#-----
# plex_flatfiles.sh
# Create plex & vol for flatfile space
#-----
NUM=1
sdNUM=1
while [ $sdNUM -le 1 ]
do

volmake plex pl_$NUM
sd=sd2_flat${sdNUM},sd3_flat${sdNUM},sd4_flat${sdNUM},sd5_flat${sdNUM},s
d6_flat${sdNUM},sd7_flat${sdNUM},sd8_flat${sdNUM},sd9_flat${sdNUM},sd10_
flat${sdNUM},sd11_flat${sdNUM},sd12_flat${sdNUM},sd13_flat${sdNUM},sd19_
flat${sdNUM},sd20_flat${sdNUM},sd21_flat${sdNUM},sd22_flat${sdNUM},sd23_
flat${sdNUM},sd24_flat${sdNUM},sd25_flat${sdNUM},sd26_flat${sdNUM},sd27_
flat${sdNUM},sd28_flat${sdNUM},sd29_flat${sdNUM},sd30_flat${sdNUM}
layout=STRIPE stwidth=$WIDTH ncol=24

volmake -U gen vol vol_$NUM plex=pl_$NUM read_pol=SELECT user=oracle
group=dba mode=0777 log_type=NONE len=0

volume start vol_$NUM
volume init active vol_$NUM
NUM=`expr $NUM + 1`
sdNUM=`expr $sdNUM + 1`
done

echo "end of plex_flatfiles.sh"

```

plex_lineitem.sh

```

#!/bin/sh
#
set -x
#
#
WIDTH=256
#
#-----
# plex_lineitem.sh
# Create plexes & vols for lineitem.
#-----
NUM=26
sdNUM=1
while [ $sdNUM -le 24 ]
do

volmake plex pl_$NUM
sd=sd2_line${sdNUM},sd3_line${sdNUM},sd4_line${sdNUM},sd5_line${sdNUM}
,sd6_line${sdNUM},sd7_line${sdNUM},sd8_line${sdNUM},sd9_line${sdNUM},sd
10_line${sdNUM},sd11_line${sdNUM},sd12_line${sdNUM},sd13_line${sdNUM},
sd19_line${sdNUM},sd20_line${sdNUM},sd21_line${sdNUM},sd22_line${sdNUM}
,sd23_line${sdNUM},sd24_line${sdNUM},sd25_line${sdNUM},sd26_line${sdNU
M},sd27_line${sdNUM},sd28_line${sdNUM},sd29_line${sdNUM},sd30_line${sdN
UM} layout=STRIPE stwidth=$WIDTH ncol=24

```

```

volmake -U gen vol vol_$NUM plex=pl_$NUM read_pol=SELECT user=oracle
group=dba mode=0777 log_type=NONE len=0

```

```

volume start vol_$NUM
volume init active vol_$NUM
NUM=`expr $NUM + 1`
sdNUM=`expr $sdNUM + 1`
done

```

```

echo "end of plex_lineitem.sh"

```

plex_orders.sh

```

#!/bin/sh
#
set -x
#
#
WIDTH=256
#
#-----
# plex_orders.sh
# Create plexes & vols for orders.
#-----
NUM=2
sdNUM=1
while [ $sdNUM -le 24 ]
do

volmake plex pl_$NUM
sd=sd2_ord${sdNUM},sd3_ord${sdNUM},sd4_ord${sdNUM},sd5_ord${sdNUM},s
d6_ord${sdNUM},sd7_ord${sdNUM},sd8_ord${sdNUM},sd9_ord${sdNUM},sd10_
ord${sdNUM},sd11_ord${sdNUM},sd12_ord${sdNUM},sd13_ord${sdNUM},sd19_
ord${sdNUM},sd20_ord${sdNUM},sd21_ord${sdNUM},sd22_ord${sdNUM},sd23_
ord${sdNUM},sd24_ord${sdNUM},sd25_ord${sdNUM},sd26_ord${sdNUM},sd27_
ord${sdNUM},sd28_ord${sdNUM},sd29_ord${sdNUM},sd30_ord${sdNUM}
layout=STRIPE stwidth=$WIDTH ncol=24

volmake -U gen vol vol_$NUM plex=pl_$NUM read_pol=SELECT user=oracle
group=dba mode=0777 log_type=NONE len=0

volume start vol_$NUM
volume init active vol_$NUM
NUM=`expr $NUM + 1`
sdNUM=`expr $sdNUM + 1`
done

echo "end of plex_orders.sh"

```

plex_temps.sh

```

#!/bin/sh
#
set -x
#
#
WIDTH=256
#
#-----
# plex_temps.sh
# Create plexes & vols for temps and ts_i_11&2 .
#-----
NUM=104
sdNUM=1
while [ $sdNUM -le 36 ]
do

volmake plex pl_$NUM
sd=sd2_temps${sdNUM},sd3_temps${sdNUM},sd4_temps${sdNUM},sd5_temps${
sdNUM},sd6_temps${sdNUM},sd7_temps${sdNUM},sd8_temps${sdNUM},sd9_te
mps${sdNUM},sd10_temps${sdNUM},sd11_temps${sdNUM},sd12_temps${sdNU
M},sd13_temps${sdNUM},sd19_temps${sdNUM},sd20_temps${sdNUM},sd21_tem
ps${sdNUM},sd22_temps${sdNUM},sd23_temps${sdNUM},sd24_temps${sdNUM}

```

```
,sd25_temps${sdNUM},sd26_temps${sdNUM},sd27_temps${sdNUM},sd28_temps${sdNUM},sd29_temps${sdNUM},sd30_temps${sdNUM} layout=STRIPE
stwidth=$WIDTH ncol=24
```

```
volmake -U gen vol vol_$NUM plex=pl_$NUM read_pol=SELECT user=oracle
group=dba mode=0777 log_type=NONE len=0
```

```
volume start vol_$NUM
volume init active vol_$NUM
NUM=`expr $NUM + 1`
sdNUM=`expr $sdNUM + 1`
done
```

```
echo "end of plex_temps.sh"
```

prepare4acid.sh

```
#!/bin/ksh
#
# $Header: prepare4acid.sh 12-aug-99.17:09:18 mpoess Exp $
#
# prepare4acid.sh
#
# Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
# NAME
#   prepare4acid.sh
#
# DESCRIPTION
#   Prepares the qualification database for the acid tests.
#
# NOTES
#
# MODIFIED (MM/DD/YY)
#   mpoess    08/12/99 - Creation
#   mpoess    08/12/99 - Creation
#
. $KIT_DIR/env

sqlplus tpced/tpcd @atrans
sqlplus tpced/tpcd @d_hist
```

create_100gb_tpch_links.sh

```
#!/bin/ksh
# Establish Links For Oracle 100GB TPC-H Database Tablespace Datafiles...
```

```
ln -s /dev/rvol/vol_1      flatfiles
ln -s /dev/rvol/vol_2      ts_o1_1
ln -s /dev/rvol/vol_3      ts_o2_1
ln -s /dev/rvol/vol_4      ts_o3_1
ln -s /dev/rvol/vol_5      ts_o4_1
ln -s /dev/rvol/vol_6      ts_o5_1
ln -s /dev/rvol/vol_7      ts_o6_1
ln -s /dev/rvol/vol_8      ts_o7_1
ln -s /dev/rvol/vol_9      ts_o8_1
ln -s /dev/rvol/vol_10     ts_o9_1
ln -s /dev/rvol/vol_11     ts_o10_1
ln -s /dev/rvol/vol_12     ts_o11_1
ln -s /dev/rvol/vol_13     ts_o12_1
ln -s /dev/rvol/vol_14     ts_o1_2
ln -s /dev/rvol/vol_15     ts_o2_2
ln -s /dev/rvol/vol_16     ts_o3_2
ln -s /dev/rvol/vol_17     ts_o4_2
ln -s /dev/rvol/vol_18     ts_o5_2
ln -s /dev/rvol/vol_19     ts_o6_2
ln -s /dev/rvol/vol_20     ts_o7_2
ln -s /dev/rvol/vol_21     ts_o8_2
ln -s /dev/rvol/vol_22     ts_o9_2
ln -s /dev/rvol/vol_23     ts_o10_2
ln -s /dev/rvol/vol_24     ts_o11_2
ln -s /dev/rvol/vol_25     ts_o12_2
ln -s /dev/rvol/vol_26     ts_i1_1
ln -s /dev/rvol/vol_27     ts_i2_1
```

```
ln -s /dev/rvol/vol_28     ts_i3_1
ln -s /dev/rvol/vol_29     ts_i4_1
ln -s /dev/rvol/vol_30     ts_i5_1
ln -s /dev/rvol/vol_31     ts_i6_1
ln -s /dev/rvol/vol_32     ts_i7_1
ln -s /dev/rvol/vol_33     ts_i8_1
ln -s /dev/rvol/vol_34     ts_i9_1
ln -s /dev/rvol/vol_35     ts_i10_1
ln -s /dev/rvol/vol_36     ts_i11_1
ln -s /dev/rvol/vol_37     ts_i12_1
ln -s /dev/rvol/vol_38     ts_i1_2
ln -s /dev/rvol/vol_39     ts_i2_2
ln -s /dev/rvol/vol_40     ts_i3_2
ln -s /dev/rvol/vol_41     ts_i4_2
ln -s /dev/rvol/vol_42     ts_i5_2
ln -s /dev/rvol/vol_43     ts_i6_2
ln -s /dev/rvol/vol_44     ts_i7_2
ln -s /dev/rvol/vol_45     ts_i8_2
ln -s /dev/rvol/vol_46     ts_i9_2
ln -s /dev/rvol/vol_47     ts_i10_2
ln -s /dev/rvol/vol_48     ts_i11_2
ln -s /dev/rvol/vol_49     ts_i12_2
ln -s /dev/rvol/vol_50     ts_s1
ln -s /dev/rvol/vol_51     ts_s2
ln -s /dev/rvol/vol_52     ts_c1
ln -s /dev/rvol/vol_53     ts_c2
ln -s /dev/rvol/vol_54     ts_p1
ln -s /dev/rvol/vol_55     ts_p2
ln -s /dev/rvol/vol_56     ts_p3
ln -s /dev/rvol/vol_57     ts_p4
ln -s /dev/rvol/vol_58     ts_ps1
ln -s /dev/rvol/vol_59     ts_ps2
ln -s /dev/rvol/vol_60     ts_ps3
ln -s /dev/rvol/vol_61     ts_ps4
ln -s /dev/rvol/vol_62     ts_ps5
ln -s /dev/rvol/vol_63     ts_ps6
ln -s /dev/rvol/vol_64     ts_ps7
ln -s /dev/rvol/vol_65     ts_ps8
ln -s /dev/rvol/vol_66     ts_i_11
ln -s /dev/rvol/vol_67     ts_i_12
ln -s /dev/rvol/vol_68     ts_i_13
ln -s /dev/rvol/vol_69     ts_i_14
ln -s /dev/rvol/vol_70     ts_i_o1
ln -s /dev/rvol/vol_71     ts_i_o2
ln -s /dev/rvol/vol_72     ts_i_o3
ln -s /dev/rvol/vol_73     ts_i_o4
ln -s /dev/rvol/vol_74     ts_i_ps1
ln -s /dev/rvol/vol_75     ts_i_ps2
ln -s /dev/rvol/vol_76     ts_i_ps3
ln -s /dev/rvol/vol_77     ts_i_ps4
ln -s /dev/rvol/vol_78     ts_i_c1
ln -s /dev/rvol/vol_79     ts_i_c2
ln -s /dev/rvol/vol_80     redolog1
ln -s /dev/rvol/vol_81     redolog2
ln -s /dev/rvol/vol_82     redolog3
ln -s /dev/rvol/vol_83     redolog4
ln -s /dev/rvol/vol_84     redolog5
ln -s /dev/rvol/vol_85     redolog6
ln -s /dev/rvol/vol_86     redolog7
ln -s /dev/rvol/vol_87     redolog8
ln -s /dev/rvol/vol_88     redolog9
ln -s /dev/rvol/vol_89     redolog10
ln -s /dev/rvol/vol_90     redolog11
ln -s /dev/rvol/vol_91     redolog12
ln -s /dev/rvol/vol_92     system1
ln -s /dev/rvol/vol_93     system2
ln -s /dev/rvol/vol_94     system3
ln -s /dev/rvol/vol_95     system4
ln -s /dev/rvol/vol_96     ts_undo1
ln -s /dev/rvol/vol_97     ts_undo2
ln -s /dev/rvol/vol_98     ts_undo3
ln -s /dev/rvol/vol_99     ts_undo4
ln -s /dev/rvol/vol_100    ts_undo5
ln -s /dev/rvol/vol_101    ts_undo6
ln -s /dev/rvol/vol_102    ts_undo7
ln -s /dev/rvol/vol_103    ts_undo8
ln -s /dev/rvol/vol_104    ts_temp1
ln -s /dev/rvol/vol_105    ts_temp2
ln -s /dev/rvol/vol_106    ts_temp3
```

```
ln -s /dev/rvol/vol_107 ts_temp4
ln -s /dev/rvol/vol_108 ts_temp5
ln -s /dev/rvol/vol_109 ts_temp6
ln -s /dev/rvol/vol_110 ts_temp7
ln -s /dev/rvol/vol_111 ts_temp8
ln -s /dev/rvol/vol_112 ts_temp9
ln -s /dev/rvol/vol_113 ts_temp10
ln -s /dev/rvol/vol_114 ts_temp11
ln -s /dev/rvol/vol_115 ts_temp12
ln -s /dev/rvol/vol_116 ts_temp13
ln -s /dev/rvol/vol_117 ts_temp14
ln -s /dev/rvol/vol_118 ts_temp15
ln -s /dev/rvol/vol_119 ts_temp16
ln -s /dev/rvol/vol_120 ts_temp17
ln -s /dev/rvol/vol_121 ts_temp18
ln -s /dev/rvol/vol_122 ts_temp19
ln -s /dev/rvol/vol_123 ts_temp20
ln -s /dev/rvol/vol_124 ts_temp21
ln -s /dev/rvol/vol_125 ts_temp22
ln -s /dev/rvol/vol_126 ts_temp23
ln -s /dev/rvol/vol_127 ts_temp24
ln -s /dev/rvol/vol_128 ts_temp25
ln -s /dev/rvol/vol_129 ts_temp26
ln -s /dev/rvol/vol_130 ts_temp27
ln -s /dev/rvol/vol_131 ts_temp28
ln -s /dev/rvol/vol_132 ts_temp29
ln -s /dev/rvol/vol_133 ts_temp30
ln -s /dev/rvol/vol_134 ts_temp31
ln -s /dev/rvol/vol_135 ts_temp32
ln -s /dev/rvol/vol_136 ts_temp33
ln -s /dev/rvol/vol_137 ts_temp34
ln -s /dev/rvol/vol_138 ts_temp35
ln -s /dev/rvol/vol_139 ts_temp36
```

Appendix C: Query Text and Result Output

mqs00q01

Begin Execution at Thu Apr 11 11:02:04 2002

-- 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 6.04 seconds.

Ended Executing this Stream at Thu Apr 11 11:02:10 2002

Stream Started at 1018540924.36
Stream Ended at 1018540930.40
Stream Processed in 6.04 seconds

SQL statements processed: 1

mqs00q02

Begin Execution at Thu Apr 11 11:02:10 2002

-- 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	
B3rqp0xbSEim4Mpy2RH J		33-320-228-2957
blithely special packages are. stealthily express deposits across the closely final		
instructi		

9923.77	Supplier#000002324	GERMANY	qNHZ7WmCzygwMPRDO9Ps	29-973-481-1831
29821.00	Manufacturer#4		furiously final deposits	
y3OD9UyWSTOk	17-779-299-1839		9558.10	Supplier#000003532 UNITED KINGDOM
quickly express packages breach quiet pinto beans. requ			88515.00	Manufacturer#4
9871.22	Supplier#000006373	GERMANY	EOeuiOn21OVpTIGuufFDFsbN1p0lhpXHp	33-152-301-2164
43868.00	Manufacturer#5		daring, sly accounts breach about th	
J8fcXWsTqM	17-813-485-8637		9492.79	Supplier#000005975 GERMANY
never silent deposits integrate furiously blit			25974.00	Manufacturer#5
9870.78	Supplier#000001286	GERMANY	S6mliCTx82z7lV	17-992-579-4839
81285.00	Manufacturer#2		always pending packages boost slyly.	
YKA,E2fjiVd7eUrzp2Ef8j1QxGo2DFnosaTEH	17-516-924-4574		9461.05	Supplier#000002536 UNITED KINGDOM
final theodolites cajole slyly special,			20033.00	Manufacturer#1
9870.78	Supplier#000001286	GERMANY	8mmGbyzaU 7ZS2wJumTibypncu9pNkDc4FYA	33-556-973-5522
181285.00	Manufacturer#4		even foxes are quickly furiously express requests. packages	
YKA,E2fjiVd7eUrzp2Ef8j1QxGo2DFnosaTEH	17-516-924-4574		9453.01	Supplier#000000802 ROMANIA
final theodolites cajole slyly special,			175767.00	Manufacturer#1
9852.52	Supplier#000008973	RUSSIA	,6HYXb4uaHITmtMBj4Ak57Pd	29-342-882-6463
18972.00	Manufacturer#2		final, regular packages across the slowly regular packag	
t5L67YdBYH6o,Vz24jpDyQ9	32-188-594-7038		9408.65	Supplier#000007772 UNITED KINGDOM
quickly regular instructions wake-- carefully unusual braids into the expres			117771.00	Manufacturer#4
9847.83	Supplier#000008097	RUSSIA	AiC5YAH,gnu0i7	33-152-491-1126
130557.00	Manufacturer#2		blithely final ideas sleep carefully. requests are	
xMe97bpE69NzdwLoX	32-375-640-3593		9359.61	Supplier#000004856 ROMANIA
slyly regular dependencies sleep slyly furiously express dep			62349.00	Manufacturer#5
9847.57	Supplier#000006345	FRANCE	HYogcF3Jb yhl	29-334-870-9731
86344.00	Manufacturer#1		carefully unusual packages sleep carefully even ideas. dogged account	
VSt3rzK3qG698u6ld8HhOByvrTcStsvQIDQDag	16-886-766-7945		9357.45	Supplier#000006188 UNITED KINGDOM
silent pinto beans should have to snooze carefully along the final reques			138648.00	Manufacturer#1
9847.57	Supplier#000006345	FRANCE	g801,ssP8wpTk4Hm	33-583-607-1633
173827.00	Manufacturer#2		carefully regular deposits wake carefully furiously even i	
VSt3rzK3qG698u6ld8HhOByvrTcStsvQIDQDag	16-886-766-7945		9352.04	Supplier#000003439 GERMANY
silent pinto beans should have to snooze carefully along the final reques			170921.00	Manufacturer#4
9836.93	Supplier#000007342	RUSSIA	qYPDgoiBGhCYxjgC	17-128-996-4650
4841.00	Manufacturer#4		fluffily regular pinto beans wake. unusual, final ideas c	
JOIK7C1,7xrEZSSOw	32-399-414-5385		9312.97	Supplier#000007807 RUSSIA
final accounts haggle. bold accounts are furiously dugouts. furiously silent asymptotes			90279.00	Manufacturer#5
are slyly			oGYMPck9XHGB2PBfKRnHA	32-673-872-5854
9817.10	Supplier#000002352	RUSSIA	unusual asymptotes above the	
124815.00	Manufacturer#2		9312.97	Supplier#000007807 RUSSIA
4LfoHUZjgEbaKw TgdKcgOc4D4uCYw	32-551-831-1437		100276.00	Manufacturer#5
blithely pending packages across the ironic accounts grow slyly after the furiously			oGYMPck9XHGB2PBfKRnHA	32-673-872-5854
9817.10	Supplier#000002352	RUSSIA	unusual asymptotes above the	
152351.00	Manufacturer#3		9280.27	Supplier#000007194 ROMANIA
4LfoHUZjgEbaKw TgdKcgOc4D4uCYw	32-551-831-1437		47193.00	Manufacturer#3
blithely pending packages across the ironic accounts grow slyly after the furiously			zhRUQkBSrFYxIAXTfInj vyGRQjeK	29-318-454-2133
9739.86	Supplier#000003384	FRANCE	slyly ironic requests despite the unusual ins	
138357.00	Manufacturer#2		9274.80	Supplier#000008854 RUSSIA
o,Z3v4POifevE k9U1b 6J1uCX,I	16-494-913-5925		76346.00	Manufacturer#3
slyly ironic theodolites hag			1xhLoOUM7I3mZ1mKnerw OSqdbb4QbGa	32-524-148-5221
9721.95	Supplier#000008757	UNITED KINGDOM	ruthlessly ironic instructions along the regular, furious requests integrate car	
156241.00	Manufacturer#3		9249.35	Supplier#000003973 FRANCE
Atg6GnM4dT2	33-821-407-2995		26466.00	Manufacturer#1
ironic, even dolphins above the furiously ironic foxes sleep slyly around the caref			d18GiDsL6Wm2lsGXM,RZf1jCsgZAOjNYVThTRP4	16-722-866-1658
9681.33	Supplier#000008406	RUSSIA	quickly ironic sauternes use b	
78405.00	Manufacturer#1		9249.35	Supplier#000003973 FRANCE
,qUuXcftUl	32-139-873-8571		33972.00	Manufacturer#1
furiously even deposits affix thinly special theodolites. furiou			d18GiDsL6Wm2lsGXM,RZf1jCsgZAOjNYVThTRP4	16-722-866-1658
9643.55	Supplier#000005148	ROMANIA	quickly ironic sauternes use b	
107617.00	Manufacturer#1		9208.70	Supplier#000007769 ROMANIA
kT4ciVFslx9z4s79p Js825	29-252-617-4850		40256.00	Manufacturer#5
doggedly even ideas boost furiously against the furiously express			rsimdze 5o9P Ht7xS	29-964-424-9649
9624.82	Supplier#000001816	FRANCE	furiously ruthless epitaphs among the furiously regular accounts use slowly fluffily ev	
34306.00	Manufacturer#3		9201.47	Supplier#000009690 UNITED KINGDOM
e7vab91vLJPWxxZnewmnDBpDmxYHrb	16-392-237-6726		67183.00	Manufacturer#5
blithely regular accounts cajole furiously. regular			CB BnUTlmi5zdeEl7R7	33-121-267-9529
9624.78	Supplier#000009658	ROMANIA	blithely unusual accounts integrate slyly. platelets	
189657.00	Manufacturer#1		9192.10	Supplier#000000115 UNITED KINGDOM
oE9uBgEfSS4opIcepXyAYM,x	29-748-876-2014		85098.00	Manufacturer#3
regular deposits haggle. furiously express asympto			nJ 2t0f7Ve,wL1,6WzGBJLNBUCKlsV	33-597-248-1220
9612.94	Supplier#000003228	ROMANIA	slyly bold pinto beans boost across the furiously regular packages. carefully regu	
120715.00	Manufacturer#2		9189.98	Supplier#000001226 GERMANY
KDdpNKN3cWu7ZSrbdp7AfSLxx,qWB	29-325-784-8187		21225.00	Manufacturer#4
carefully pending accounts serve. furiously close deposits boost slyly. q			qsLCqSvLyZfuXlpjz	17-725-903-1381
9612.94	Supplier#000003228	ROMANIA	final, express instruction	
198189.00	Manufacturer#4		9128.97	Supplier#000004311 RUSSIA
KDdpNKN3cWu7ZSrbdp7AfSLxx,qWB	29-325-784-8187		146768.00	Manufacturer#5
carefully pending accounts serve. furiously close deposits boost slyly. q			I8IjnXd7NSJRs594RxsRR0	32-155-440-7120
9571.83	Supplier#000004305	ROMANIA	regular pinto beans sleep ca	
179270.00	Manufacturer#2		9104.83	Supplier#000008520 GERMANY

150974.00 Manufacturer#4
RqRVDgD0ER J9 b41vR2,3 17-728-804-1793
deposits sleep carefully e
9101.00 Supplier#000005791 ROMANIA
128254.00 Manufacturer#5
zub2zCV,jhHPPQqi,P2INAJE1zI n66cOEoXFG 29-549-251-5384
carefully ironic packages after the
9094.57 Supplier#000004582 RUSSIA
39575.00 Manufacturer#1
WB0XkCSG3r,mnQ n,h9VlXjtr9ARHFvKgMDf 32-587-577-1351
asymptotes above the slyly even requests haggle furiously about the regular accounts
8996.87 Supplier#000004702 FRANCE
102191.00 Manufacturer#5
8XVcQK23akp 16-811-269-8946
stealthy requests haggle c
8996.14 Supplier#000009814 ROMANIA
139813.00 Manufacturer#2
af0O5pg83IPU4IDVmEylXZVqYZQzSDIYLAmR 29-995-571-8781
ironic theodolites are evenly unusual requests-- pending pinto beans across the in
8968.42 Supplier#000010000 ROMANIA
119999.00 Manufacturer#5
aTGLEusCiL4F PDBdv665XBjHPyCOB0i 29-578-432-2146
furiously final ideas believe furiously. furiously final ideas
8936.82 Supplier#000007043 UNITED KINGDOM
109512.00 Manufacturer#1
FVajceZInZdbJE6Z9XsRUxrUEpiwHDR0Xi,1Rz 33-784-177-8208
furiously regular excuses wake after the blithely special pinto beans? even instructions
sl
8929.42 Supplier#000008770 FRANCE
173735.00 Manufacturer#4
R7cG26TtXrHAP9 HckhfRi 16-242-746-9248
final accounts sleep furiously. blithely ironic foxes wake boldly across the furiously s
8920.59 Supplier#000003967 ROMANIA
26460.00 Manufacturer#1
eHoAXe62SY9 29-194-731-3944
quickly even requests should have to affix blithely-- fur
8920.59 Supplier#000003967 ROMANIA
173966.00 Manufacturer#2
eHoAXe62SY9 29-194-731-3944
quickly even requests should have to affix blithely-- fur
8913.96 Supplier#000004603 UNITED KINGDOM
137063.00 Manufacturer#2
OUzlvMUr7n,utLxmPNeYKSf3T24OXskxB5 33-789-255-7342
slyly ironic packages detect furious accounts. ironic de
8877.82 Supplier#000007967 FRANCE
167966.00 Manufacturer#5
A3pi1BARM4nx6R,qrwFoRPU 16-442-147-9345
final deposits after the silent deposits ha
8862.24 Supplier#000003323 ROMANIA
73322.00 Manufacturer#3
W9lYcsC9FwBqk3ItL 29-736-951-3710
unusual, pending theodolites integrate furiously slyly even pinto beans. unusual
sheaves sleep before
8841.59 Supplier#000005750 ROMANIA
100729.00 Manufacturer#5
ErX3lAgu0g62iaHF9x50uMH4EgeN9HEG 29-344-502-5481
excuses after the blithely regular packages mold carefully deposits. regular a
8781.71 Supplier#000003121 ROMANIA
13120.00 Manufacturer#5
wNqTogx238ZYCamFb,50v,bj 4IbNFW9Bvw1xP 29-707-291-5144
packages are quickly after the final, even packages. furiously regular
8754.24 Supplier#000009407 UNITED KINGDOM
179406.00 Manufacturer#4
CHRCbkaWcf5B 33-903-970-9604
regular dependencies haggle across the carefully bold
8691.06 Supplier#000004429 UNITED KINGDOM
126892.00 Manufacturer#2
k,BQms5UhoAF1B2Asi,fLib 33-964-337-5038
quickly special foxes against the furiously silent platelets wake quickly after t
8655.99 Supplier#000006330 RUSSIA
193810.00 Manufacturer#2
UozlaENr0ytKe2w6CeIEWFWn iO3S8Rae7Ou 32-561-198-3705
blithely even packages alongside
8638.36 Supplier#000002920 RUSSIA
75398.00 Manufacturer#1
Je2a8bszf3L 32-122-621-7549
express deposits wake. furiously silent requests wake carefully silent instru
8638.36 Supplier#000002920 RUSSIA
170402.00 Manufacturer#3

Je2a8bszf3L 32-122-621-7549
express deposits wake. furiously silent requests wake carefully silent instru
8607.69 Supplier#000006003 UNITED KINGDOM
76002.00 Manufacturer#2
EH9wADcEiuenM0NR08zDwMidw,52Y2RyIEiA 33-416-807-5206
always special foxes wake slyly bold, ironic accounts. ironic instructions affix careful
8569.52 Supplier#000005936 RUSSIA
5935.00 Manufacturer#5
jXaNZ6vwnEWJ2ksLZJpjtgt0bY2a3AU 32-644-251-7916
packages sleep furiously. special requests about the fluffily even accounts detect
8564.12 Supplier#000000033 GERMANY
110032.00 Manufacturer#1
gfeKpYw3400L0SDyWXA6Ya1Qmq1w6YB9f3R 17-138-897-9374
ironic instructions are. special pearls above
8553.82 Supplier#000003979 ROMANIA
143978.00 Manufacturer#4
BfmVhCAncMY3jzpjUMy4CNWs9 HzpdQR7INJU 29-124-646-4897
express, ironic pinto beans cajole around the express, even packages. qu
8517.23 Supplier#000009529 RUSSIA
37025.00 Manufacturer#5
e44R8o7JAIS9iMcr 32-565-297-8775
furiously silent requests cajole furiously furiously ironic foxes. slyly express p
8517.23 Supplier#000009529 RUSSIA
59528.00 Manufacturer#2
e44R8o7JAIS9iMcr 32-565-297-8775
furiously silent requests cajole furiously furiously ironic foxes. slyly express p
8503.70 Supplier#000006830 RUSSIA
44325.00 Manufacturer#4
BC4WFCYRUZyaIgchU 4S 32-147-878-5069
quickly regular excuses detect evenly around
8457.09 Supplier#000009456 UNITED KINGDOM
19455.00 Manufacturer#1
7SBhZs8gP1cJtOQf433YBk 33-858-440-4349
carefully final accounts sleep blithely special foxes. slyly regular pinto beans alon
8441.40 Supplier#000003817 FRANCE
141302.00 Manufacturer#2
hU3fz3xL78 16-339-356-5115
blithely blithe ideas are
8432.89 Supplier#000003990 RUSSIA
191470.00 Manufacturer#1
webHBp1RQbfxAYDASS75MsywmsKHRVdkrvNe6m 32-839-509-9301
final requests along the blithely ironic packages kindle against the carefully fina
8431.40 Supplier#000002675 ROMANIA
5174.00 Manufacturer#1
HJFStOu9R5NGPOegKhgbzBdyvrG2yh8w 29-474-643-1443
express, final deposits cajole carefully. stealthily unusual requests
8407.04 Supplier#000005406 RUSSIA
162889.00 Manufacturer#4
j7 gYF5RW8DC5UrjKC 32-626-152-4621
quickly final sheaves boost. car
8386.08 Supplier#000008518 FRANCE
36014.00 Manufacturer#3
2jqzqqAVe9crMVGp,n9nTsQXuNLtUYoJEDcqvWV 16-618-780-7481
slyly ironic theodolites are slyly. dogged, pendin
8376.52 Supplier#000005306 UNITED KINGDOM
190267.00 Manufacturer#5
9t8Y8 QqSisoADPt6NLdk,TP5zyRx41oBUlgoGc9 33-632-514-7931
furiously even instructions integrate during the furiously regular re
8348.74 Supplier#000008851 FRANCE
66344.00 Manufacturer#4
nWxi7GwEbhw1 16-796-240-2472
ironic instructions nag slyly against the slyly even theodolites. requests alongside of
8338.58 Supplier#000007269 FRANCE
17268.00 Manufacturer#4
ZwhJSwABUoiB04,3 16-267-277-4365
ruthlessly regular asymptotes a
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
dqblvV8dCNAorGIJ 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
 wgfsorVPexl9pEXWywaqlBMDYYf 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 CtXS2S16i 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
 Dh8Ikg39onrbOL4DyTfGw8a9oKUX3d9Y 32-836-132-8872
 carefully regular instructions integrate blithely silent foxes. furiously express
 instructions hagg
 8042.09 Supplier#000003245 RUSSIA
 150729.00 Manufacturer#1
 Dh8Ikg39onrbOL4DyTfGw8a9oKUX3d9Y 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
 kkYvL6luvojJgTNG IKKaXQDYgx8llohj 17-627-663-8014
 quickly regular requests are furiously. pending deposits wake
 7937.93 Supplier#000009012 ROMANIA
 83995.00 Manufacturer#2
 iUiTziH,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,v03IKzfTd,1PYsFXQFFOG 17-266-947-7315
 final requests integrate slyly above the silent, even
 7912.91 Supplier#000004211 GERMANY
 184210.00 Manufacturer#4
 2wQRVovHrm3,v03IKzfTd,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
 Y28ITVeYriT3kiGdV2K8fSZ V2UqT5HlOtz 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
 2Z0JGkiv01Y00oCFwUGfviIbhzcDy 16-464-517-8943
 carefully bold accounts doub

100 rows processed.
 Query Processed in 1.32 seconds.

Ended Executing this Stream at Thu Apr 11 11:02:11 2002

Stream Started at 1018540930.57
 Stream Ended at 1018540931.89
 Stream Processed in 1.32 seconds

SQL statements processed: 1

mqs00q03

Begin Execution at Thu Apr 11 11:02:12 2002

-- 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 2.33 seconds.

Ended Executing this Stream at Thu Apr 11 11:02:14 2002

Stream Started at 1018540932.05
 Stream Ended at 1018540934.37
 Stream Processed in 2.33 seconds

SQL statements processed: 1

mqs00q04

Begin Execution at Thu Apr 11 11:02:14 2002

-- 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 2.41 seconds.

Ended Executing this Stream at Thu Apr 11 11:02:16 2002

Stream Started at 1018540934.53
Stream Ended at 1018540936.94
Stream Processed in 2.41 seconds

SQL statements processed: 1

mqs00q05

Begin Execution at Thu Apr 11 11:02:17 2002

-- 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 3.81 seconds.

Ended Executing this Stream at Thu Apr 11 11:02:20 2002

Stream Started at 1018540937.09
Stream Ended at 1018540940.90
Stream Processed in 3.81 seconds

SQL statements processed: 1

mqs00q06

Begin Execution at Thu Apr 11 11:02:21 2002

-- 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 0.80 seconds.

Ended Executing this Stream at Thu Apr 11 11:02:21 2002

Stream Started at 1018540941.06
Stream Ended at 1018540941.86
Stream Processed in 0.80 seconds

SQL statements processed: 1

mqs00q07

Begin Execution at Thu Apr 11 11:02:22 2002

-- 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 2.10 seconds.

Ended Executing this Stream at Thu Apr 11 11:02:24 2002

Stream Started at 1018540942.02
Stream Ended at 1018540944.12
Stream Processed in 2.10 seconds

SQL statements processed: 1

mqs00q08

Begin Execution at Thu Apr 11 11:02:24 2002

-- 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 6.17 seconds.

Ended Executing this Stream at Thu Apr 11 11:02:30 2002

Stream Started at 1018540944.28
Stream Ended at 1018540950.45
Stream Processed in 6.17 seconds

SQL statements processed: 1

mqs00q09

Begin Execution at Thu Apr 11 11:02:30 2002

-- using default substitutions

```

select
nation,
o_year,
sum(amount) as sum_profit
from
(
select
n_name as nation,
to_number (to_char (o_orderdate, 'yyyy')) as o_year,
l_extendedprice * (1 - l_discount) - ps_supplycost * l_quantity as amount
from
part,
supplier,
lineitem,

```

```

partsupp,
orders,

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
CANADA	1998.00	30379833.77
CANADA	1997.00	50465052.31
CANADA	1996.00	52560501.39
CANADA	1995.00	52375332.81
CANADA	1994.00	52600364.66
CANADA	1993.00	52644504.07
CANADA	1992.00	53932871.70
CHINA	1998.00	31075466.16
CHINA	1997.00	50551874.45
CHINA	1996.00	51039293.88
CHINA	1995.00	49287534.62
CHINA	1994.00	50851090.07
CHINA	1993.00	54229629.83
CHINA	1992.00	52400529.37
EGYPT	1998.00	29054433.39
EGYPT	1997.00	50627611.45
EGYPT	1996.00	49542212.84
EGYPT	1995.00	48311550.32
EGYPT	1994.00	49790644.74
EGYPT	1993.00	48904292.97
EGYPT	1992.00	49434932.62
ETHIOPIA	1998.00	28040717.27
ETHIOPIA	1997.00	47455009.87
ETHIOPIA	1996.00	46491097.57
ETHIOPIA	1995.00	46804449.30
ETHIOPIA	1994.00	48516143.92
ETHIOPIA	1993.00	46551891.56
ETHIOPIA	1992.00	44934648.64
FRANCE	1998.00	32226407.84
FRANCE	1997.00	47121485.86
FRANCE	1996.00	47263135.50
FRANCE	1995.00	47275997.57
FRANCE	1994.00	47067209.33
FRANCE	1993.00	51163370.11
FRANCE	1992.00	47846235.33
GERMANY	1998.00	28624942.66
GERMANY	1997.00	49309074.88
GERMANY	1996.00	49918683.17

GERMANY	1995.00	52650718.72
GERMANY	1994.00	50346900.42
GERMANY	1993.00	50991895.81
GERMANY	1992.00	48274126.10
INDIA	1998.00	29943144.35
INDIA	1997.00	50665453.23
INDIA	1996.00	50283092.29
INDIA	1995.00	50006774.64
INDIA	1994.00	48995190.76
INDIA	1993.00	50286902.85
INDIA	1992.00	50850329.40
INDONESIA	1998.00	27672340.00
INDONESIA	1997.00	50512145.73
INDONESIA	1996.00	51653060.12
INDONESIA	1995.00	51508779.59
INDONESIA	1994.00	52817950.32
INDONESIA	1993.00	47959994.96
INDONESIA	1992.00	51776605.03
IRAN	1998.00	29065736.24
IRAN	1997.00	50042063.05
IRAN	1996.00	50926653.19
IRAN	1995.00	51249667.65
IRAN	1994.00	50337085.87
IRAN	1993.00	51730763.49
IRAN	1992.00	49955856.56
IRAQ	1998.00	31624551.00
IRAQ	1997.00	55121749.02
IRAQ	1996.00	55897663.79
IRAQ	1995.00	54815472.52
IRAQ	1994.00	54408516.13
IRAQ	1993.00	53633167.98
IRAQ	1992.00	55891939.34
JAPAN	1998.00	27934179.67
JAPAN	1997.00	44517162.55
JAPAN	1996.00	42545606.12
JAPAN	1995.00	43749356.40
JAPAN	1994.00	44840243.07
JAPAN	1993.00	44660015.53
JAPAN	1992.00	45410249.12
JORDAN	1998.00	26901488.58
JORDAN	1997.00	45471878.41
JORDAN	1996.00	46794325.79
JORDAN	1995.00	45178828.58
JORDAN	1994.00	45333636.51
JORDAN	1993.00	47971496.10
JORDAN	1992.00	44717239.18
KENYA	1998.00	28597614.34
KENYA	1997.00	47949733.73
KENYA	1996.00	46886924.62
KENYA	1995.00	46072338.76
KENYA	1994.00	45772061.17
KENYA	1993.00	46308728.23
KENYA	1992.00	47257780.84
MOROCCO	1998.00	26732115.58
MOROCCO	1997.00	45637304.25
MOROCCO	1996.00	45558221.75
MOROCCO	1995.00	47851318.89
MOROCCO	1994.00	46272172.94
MOROCCO	1993.00	46764326.18
MOROCCO	1992.00	48122783.58
MOZAMBIQUE	1998.00	30712392.01
MOZAMBIQUE	1997.00	50316528.76
MOZAMBIQUE	1996.00	51640320.25
MOZAMBIQUE	1995.00	50693774.51
MOZAMBIQUE	1994.00	49253277.63
MOZAMBIQUE	1993.00	49153016.54
MOZAMBIQUE	1992.00	48247551.85
PERU	1998.00	29326102.32
PERU	1997.00	49753780.40
PERU	1996.00	50935170.29
PERU	1995.00	53309883.41
PERU	1994.00	50643531.80
PERU	1993.00	51584622.00
PERU	1992.00	47523899.05
ROMANIA	1998.00	30368667.40
ROMANIA	1997.00	50365683.85
ROMANIA	1996.00	49598999.01
ROMANIA	1995.00	47537642.87
ROMANIA	1994.00	51455283.01

ROMANIA	1993.00	50407136.89
ROMANIA	1992.00	48185385.13
RUSSIA	1998.00	28322384.03
RUSSIA	1997.00	50106685.18
RUSSIA	1996.00	51753342.43
RUSSIA	1995.00	49215820.36
RUSSIA	1994.00	52205666.44
RUSSIA	1993.00	51860230.03
RUSSIA	1992.00	53251677.15
SAUDI ARABIA	1998.00	31541259.81
SAUDI ARABIA	1997.00	52438750.81
SAUDI ARABIA	1996.00	52543737.82
SAUDI ARABIA	1995.00	52938696.53
SAUDI ARABIA	1994.00	51389601.97
SAUDI ARABIA	1993.00	52937508.88
SAUDI ARABIA	1992.00	54843459.64
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 5.92 seconds.

Ended Executing this Stream at Thu Apr 11 11:02:36 2002

Stream Started at 1018540950.61
Stream Ended at 1018540956.53
Stream Processed in 5.92 seconds

SQL statements processed: 1

mqs00q10

Begin Execution at Thu Apr 11 11:02:36 2002

-- using default substitutions

```
select * from (
select
c_custkey,
c_name,
sum(l_extendedprice * (1 - l_discount)) as revenue,
c_acctbal,
n_name,
c_address,
c_phone,
c_comment
from
customer,
orders,
lineitem,
nation
where
```

```
c_custkey = o_custkey
and l_orderkey = o_orderkey
and o_orderdate >= to_date ('1993-10-01', 'YYYY-MM-DD')
and o_orderdate < add_months( to_date( '1993-10-01', 'YYYY-MM-DD'), 3)
and l_returnflag = 'R'
and c_nationkey = n_nationkey
group by
c_custkey,
c_name,
c_acctbal,
c_phone,
n_name,
c_address,
c_comment
order by
revenue desc)
where rownum <= 20
```

C_CUSTKEY	C_NAME	REVENUE
C_ACCTBAL	N_NAME	
C_ADDRESS	C_PHONE	
C_COMMENT		
57040.00	Customer#000057040	734235.25
632.87	JAPAN	
Eioyzjf4pp	22-895-641-3466	
requests sleep blithely about the furiously i		
143347.00	Customer#000143347	721002.69
2557.47	EGYPT	
1aReFYv,Kw4	14-742-935-3718	
fluffily bold excuses haggle finally after the u		
60838.00	Customer#000060838	679127.31
2454.77	BRAZIL	
64EaJ5vMAHWJIBOXJklpNc2RjiWE	12-913-494-9813	
furiously even pinto beans integrate under the ruthless foxes; ironic, even dolphins		
across the syl		
101998.00	Customer#000101998	637029.57
3790.89	UNITED KINGDOM	
01c9CILnNtfOQYmZj	33-593-865-6378	
accounts doze blithely! enticing, final deposits sleep blithely special accounts. slyly		
express accounts pla		
125341.00	Customer#000125341	633508.09
4983.51	GERMANY	
S29ODD6bceU8QSuueJznkNaK	17-582-695-5962	
quickly express requests wake quickly blithely		
25501.00	Customer#000025501	620269.78
7725.04	ETHIOPIA	
W556MXuoiaYCCZamJI,Rn0B4ACUGdkQ8DZ	15-874-808-6793	
quickly special requests sleep evenly among the special deposits. special deposi		
115831.00	Customer#000115831	596423.87
5098.10	FRANCE	
rFeBbEEyk dl ne7zV5fDrmiq1oK09wV7pxqCgIc	16-715-386-3788	
carefully bold excuses sleep alongside of the thinly idle		
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#00006226	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	
iANyZHjqhyy7Ajah0pTrYyhJ	10-886-956-3143	
furiously even accounts are blithely above the furious!		
115640.00	Customer#000115640	569341.19

6436.10 ARGENTINA
 Vtgfia9qI 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
 ChqEoK43OysjdHbtKCp6dKqjNyvv9 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
 HgiV0phqhaIa9aydNoIlb 29-915-458-2654
 instructions nag quickly. furiously bold accounts cajol

20 rows processed.
 Query Processed in 2.88 seconds.

Ended Executing this Stream at Thu Apr 11 11:02:39 2002

Stream Started at 1018540956.69
 Stream Ended at 1018540959.57
 Stream Processed in 2.88 seconds

SQL statements processed: 1

mq500q11

Begin Execution at Thu Apr 11 11:02:39 2002

-- using default substitutions

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

)
 order by
 value desc

PS_PARTKEY	VALUE
129760.00	17538456.86
166726.00	16503353.92
191287.00	16474801.97
161758.00	16101755.54
34452.00	15983844.72
139035.00	15907078.34
9403.00	15451755.62
154358.00	15212937.88
38823.00	15064802.86
85606.00	15053957.15
33354.00	14408297.40
154747.00	14407580.68
82865.00	14235489.78
76094.00	14094247.04
222.00	13937777.74
121271.00	13908336.00
55221.00	13716120.47
22819.00	13666434.28
76281.00	13646853.68
85298.00	13581154.93
85158.00	13554904.00
139684.00	13535538.72
31034.00	13498025.25
87305.00	13482847.04
10181.00	13445148.75
62323.00	13411824.30
26489.00	13377256.38
96493.00	13339057.83
56548.00	13329014.97
55576.00	13306843.35
159751.00	13306614.48
92406.00	13287414.50
182636.00	13223726.74
199969.00	13135288.21
62865.00	13001926.94
7284.00	12945298.19
197867.00	12944510.52
11562.00	12931575.51
75165.00	12916918.12
97175.00	12911283.50
140840.00	12896562.23
65241.00	12890600.46
166120.00	12876927.22
9035.00	12863828.70
144616.00	12853549.30
176723.00	12832309.74
170884.00	12792136.58
29790.00	12723300.33
95213.00	12555483.73
183873.00	12550533.05
171235.00	12476538.30
21533.00	12437821.32
17290.00	12432159.50
156397.00	12260623.50
122611.00	12222812.98
139155.00	12220319.25
146316.00	12215800.61
171381.00	12199734.52
198633.00	12078226.95
167417.00	12046637.62
59512.00	12043468.76
31688.00	12034893.64
159586.00	12001505.84
*** Deleted Rows Here ***	
80408.00	7963312.17
37728.00	7961875.68
26624.00	7961772.31
44736.00	7961144.10
29763.00	7960605.03
36147.00	7959463.68
146040.00	7957587.66
115469.00	7957485.14
142276.00	7956790.63
181280.00	7954037.35

115096.00	7953047.55
109650.00	7952258.73
93862.00	7951992.24
158325.00	7950728.30
55952.00	7950387.06
122397.00	7947106.27
28114.00	7946945.72
11966.00	7945197.48
47814.00	7944083.00
85096.00	7943691.06
51657.00	7943593.77
196680.00	7943578.89
13141.00	7942730.34
193327.00	7941036.25
152612.00	7940663.71
139680.00	7939242.36
31134.00	7938318.30
45636.00	7937240.85
56694.00	7936015.95
8114.00	7933921.88
71518.00	7932261.69
72922.00	7930400.64
146699.00	7929167.40
92387.00	7928972.67
186289.00	7928786.19
95952.00	7927972.78
196514.00	7927180.70
4403.00	7925729.04
2267.00	7925649.37
45924.00	7925047.68
11493.00	7916722.23
104478.00	7916253.60
166794.00	7913842.00
161995.00	7910874.27
23538.00	7909752.06
41093.00	7909579.92
112073.00	7908617.57
92814.00	7908262.50
88919.00	7907992.50
79753.00	7907933.88
108765.00	7905338.98
146530.00	7905336.60
71475.00	7903367.58
36289.00	7901946.50
61739.00	7900794.00
52338.00	7898638.08
194299.00	7898421.24
105235.00	7897829.94
77207.00	7897752.72
96712.00	7897575.27
10157.00	7897046.25
171154.00	7896814.50
79373.00	7896186.00
113808.00	7893353.88
27901.00	7892952.00
128820.00	7892882.72
25891.00	7890511.20
122819.00	7888881.02
154731.00	7888301.33
101674.00	7879324.60
51968.00	7879102.21
72073.00	7877736.11
5182.00	7874521.73

1048 rows processed.
Query Processed in 2.00 seconds.

Ended Executing this Stream at Thu Apr 11 11:02:41 2002

Stream Started at 1018540959.73
Stream Ended at 1018540961.73
Stream Processed in 2.00 seconds

SQL statements processed: 1

mqs00q12

Begin Execution at Thu Apr 11 11:02:41 2002

-- 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 2.15 seconds.

Ended Executing this Stream at Thu Apr 11 11:02:44 2002

Stream Started at 1018540961.89
Stream Ended at 1018540964.04
Stream Processed in 2.15 seconds

SQL statements processed: 1

mqs00q13

Begin Execution at Thu Apr 11 11:02:44 2002

-- 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 3.95 seconds.

Ended Executing this Stream at Thu Apr 11 11:02:48 2002

Stream Started at 1018540964.19
Stream Ended at 1018540968.14
Stream Processed in 3.95 seconds

SQL statements processed: 1

mqs00q14

Begin Execution at Thu Apr 11 11:02:48 2002

-- 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 0.32 seconds.

Ended Executing this Stream at Thu Apr 11 11:02:48 2002

Stream Started at 1018540968.30
Stream Ended at 1018540968.62
Stream Processed in 0.32 seconds

SQL statements processed: 1

mqs00q15

Begin Execution at Thu Apr 11 11:02:48 2002

-- using default substitutions

```

create view revenue0 (supplier_no, total_revenue) as
select
    l_suppkey,
    sum(l_extendedprice * (1 - l_discount))
from
    lineitem
where
    l_shipdate >= to_date('1996-01-01', 'YYYY-MM-DD')
    and l_shipdate < add_months(to_date('1996-01-01', 'YYYY-MM-DD'), 3)
group by
    l_suppkey
Query Processed in 0.07 seconds.

```

```

select
    s_suppkey,
    s_name,
    s_address,
    s_phone,
    total_revenue
from
    supplier,
    revenue0
where
    s_suppkey = supplier_no
    and total_revenue = (
        select
            max(total_revenue)
        from
            revenue0
        )
order by
    s_suppkey

```

S_SUPPKEY	S_NAME	S_PHONE	TOTAL_REVENUE
8449.00	Supplier#000008449		
Wp34zim9qYFbVctdW		20-469-856-8873	1772627.21

1 row processed.
Query Processed in 1.94 seconds.

drop view revenue0
Query Processed in 0.03 seconds.

Ended Executing this Stream at Thu Apr 11 11:02:50 2002

Stream Started at 1018540968.77
Stream Ended at 1018540970.81
Stream Processed in 2.04 seconds

SQL statements processed: 3

mq500q16

Begin Execution at Thu Apr 11 11:02:50 2002

-- using default substitutions

```
select
p_brand,
p_type,
p_size,
count(distinct ps_suppkey) as supplier_cnt
from
partsupp,
part
where
p_partkey = ps_partkey
and p_brand <> 'Brand#45'
and p_type not like 'MEDIUM POLISHED%'
and p_size in (49, 14, 23, 45, 19, 3, 36, 9)
and ps_suppkey not in (
select
s_suppkey
from
supplier
where
s_comment like '%Customer%Complaints%'
)
group by
p_brand,
p_type,
p_size
order by
supplier_cnt desc,
p_brand,
p_type,
p_size
```

P_BRAND	P_TYPE	P_SIZE	SUPPLIER_CNT
Brand#41	MEDIUM BRUSHED TIN	3.00	28.00
Brand#54	STANDARD BRUSHED COPPER	14.00	27.00
Brand#11	STANDARD BRUSHED TIN	23.00	24.00
Brand#11	STANDARD BURNISHED BRASS	36.00	24.00
Brand#15	MEDIUM ANODIZED NICKEL	3.00	24.00
Brand#15	SMALL ANODIZED BRASS	45.00	24.00
Brand#15	SMALL BURNISHED NICKEL	19.00	24.00
Brand#21	MEDIUM ANODIZED COPPER	3.00	24.00
Brand#22	SMALL BRUSHED NICKEL	3.00	24.00
Brand#22	SMALL BURNISHED BRASS	19.00	24.00
Brand#25	MEDIUM BURNISHED COPPER	36.00	24.00

Brand#31	PROMO POLISHED COPPER	36.00	24.00
Brand#33	LARGE POLISHED TIN	23.00	24.00
Brand#33	PROMO POLISHED STEEL	14.00	24.00
Brand#35	PROMO BRUSHED NICKEL	14.00	24.00
Brand#41	ECONOMY BRUSHED STEEL	9.00	24.00
Brand#41	ECONOMY POLISHED TIN	19.00	24.00
Brand#41	LARGE PLATED COPPER	36.00	24.00
Brand#42	ECONOMY PLATED BRASS	3.00	24.00
Brand#42	STANDARD POLISHED TIN	49.00	24.00
Brand#43	PROMO BRUSHED TIN	3.00	24.00
Brand#43	SMALL ANODIZED COPPER	36.00	24.00
Brand#44	STANDARD POLISHED NICKEL	3.00	24.00
Brand#52	ECONOMY PLATED TIN	14.00	24.00
Brand#52	STANDARD BURNISHED NICKEL	3.00	24.00
Brand#53	MEDIUM ANODIZED STEEL	14.00	24.00
Brand#14	PROMO ANODIZED NICKEL	45.00	23.00
Brand#32	ECONOMY PLATED BRASS	9.00	23.00
Brand#52	SMALL ANODIZED COPPER	3.00	23.00
Brand#11	ECONOMY BRUSHED COPPER	45.00	20.00
Brand#11	ECONOMY PLATED BRASS	23.00	20.00

*** Deleted Rows Here ***

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

Brand#51	SMALL PLATED BRASS	23.00	3.00
Brand#52	MEDIUM BRUSHED BRASS	45.00	3.00
Brand#53	MEDIUM BRUSHED TIN	45.00	3.00
Brand#54	ECONOMY POLISHED BRASS	9.00	3.00
Brand#55	PROMO PLATED BRASS	19.00	3.00
Brand#55	STANDARD PLATED TIN	49.00	3.00

18314 rows processed.
Query Processed in 4.69 seconds.

Ended Executing this Stream at Thu Apr 11 11:02:55 2002

Stream Started at 1018540970.96
Stream Ended at 1018540975.66
Stream Processed in 4.69 seconds

SQL statements processed: 1

mq500q17

Begin Execution at Thu Apr 11 11:02:55 2002

-- using default substitutions

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

AVG_YEARLY
348406.05

1 row_ processed.
Query Processed in 6.54 seconds.

Ended Executing this Stream at Thu Apr 11 11:03:02 2002

Stream Started at 1018540975.82
Stream Ended at 1018540982.36
Stream Processed in 6.54 seconds

SQL statements processed: 1

mq500q18

Begin Execution at Thu Apr 11 11:03:02 2002

-- using default substitutions

select * from (

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

C_NAME	C_CUSTKEY	O_ORDERKEY	O_ORDERDATE	O_TOTALPRICE	SUM(L_QUANTITY)
Customer#000128120	128120.00	4722021.00	1994-04-07	544089.09	323.00
Customer#000144617	144617.00	3043270.00	1997-02-12	530604.44	317.00
Customer#000013940	13940.00	2232932.00	1997-04-13	522720.61	304.00
Customer#000066790	66790.00	2199712.00	1996-09-30	515531.82	327.00
Customer#000046435	46435.00	4745607.00	1997-07-03	508047.99	309.00
Customer#000015272	15272.00	3883783.00	1993-07-28	500241.33	302.00
Customer#000146608	146608.00	3342468.00	1994-06-12	499794.58	303.00
Customer#000096103	96103.00	5984582.00	1992-03-16	494398.79	312.00
Customer#000024341	24341.00	1474818.00	1992-11-15	491348.26	302.00
Customer#000137446	137446.00	5489475.00	1997-05-23	487763.25	311.00
Customer#000107590	107590.00	4267751.00	1994-11-04	485141.38	301.00
Customer#000050008	50008.00	2366755.00	1996-12-09	483891.26	302.00
Customer#000015619	15619.00	3767271.00	1996-08-07	480083.96	318.00
Customer#000077260	77260.00	1436544.00	1992-09-12	479499.43	307.00
Customer#000109379	109379.00	5746311.00	1996-10-10	478064.11	302.00
Customer#000054602	54602.00	5832321.00	1997-02-09	471220.08	307.00
Customer#000105995	105995.00	2096705.00	1994-07-03	469692.58	307.00
Customer#000148885	148885.00	2942469.00	1992-05-31	469630.44	313.00
Customer#000114586	114586.00	551136.00	1993-05-19	469605.59	308.00
Customer#000105260	105260.00	5296167.00	1996-09-06	469360.57	303.00
Customer#000147197	147197.00	1263015.00	1997-02-02		

467149.67	320.00		
Customer#000064483	64483.00	2745894.00	1996-07-04
466991.35	304.00		
Customer#000136573	136573.00	2761378.00	1996-05-31
461282.73	301.00		
Customer#000016384	16384.00	502886.00	1994-04-12
458378.92	312.00		
Customer#000117919	117919.00	2869152.00	1996-06-20
456815.92	317.00		
Customer#000012251	12251.00	735366.00	1993-11-24
455107.26	309.00		
Customer#000120098	120098.00	1971680.00	1995-06-14
453451.23	308.00		
Customer#000066098	66098.00	5007490.00	1992-08-07
453436.16	304.00		
Customer#000117076	117076.00	4290656.00	1997-02-05
449545.85	301.00		
Customer#000129379	129379.00	4720454.00	1997-06-07
448665.79	303.00		
Customer#000126865	126865.00	4702759.00	1994-11-07
447606.65	320.00		
Customer#000088876	88876.00	983201.00	1993-12-30
446717.46	304.00		
Customer#000036619	36619.00	4806726.00	1995-01-17
446704.09	328.00		
Customer#000141823	141823.00	2806245.00	1996-12-29
446269.12	310.00		
Customer#000053029	53029.00	2662214.00	1993-08-13
446144.49	302.00		
Customer#000018188	18188.00	3037414.00	1995-01-25
443807.22	308.00		
Customer#000066533	66533.00	29158.00	1995-10-21
443576.50	305.00		
Customer#000037729	37729.00	4134341.00	1995-06-29
441082.97	309.00		
Customer#000003566	3566.00	2329187.00	1998-01-04
439803.36	304.00		
Customer#000045538	45538.00	4527553.00	1994-05-22
436275.31	305.00		
Customer#000081581	81581.00	4739650.00	1995-11-04
435405.90	305.00		
Customer#000119989	119989.00	1544643.00	1997-09-20
434568.25	320.00		
Customer#000003680	3680.00	3861123.00	1998-07-03
433525.97	301.00		
Customer#000113131	113131.00	967334.00	1995-12-15
432957.75	301.00		
Customer#000141098	141098.00	565574.00	1995-09-24
430986.69	301.00		
Customer#000093392	93392.00	5200102.00	1997-01-22
425487.51	304.00		
Customer#000015631	15631.00	1845057.00	1994-05-12
419879.59	302.00		
Customer#000112987	112987.00	4439686.00	1996-09-17
418161.49	305.00		
Customer#000012599	12599.00	4259524.00	1998-02-12
415200.61	304.00		
Customer#000105410	105410.00	4478371.00	1996-03-05
412754.51	302.00		
Customer#000149842	149842.00	5156581.00	1994-05-30
411329.35	302.00		
Customer#000010129	10129.00	5849444.00	1994-03-21
409129.85	309.00		
Customer#000069904	69904.00	1742403.00	1996-10-19
408513.00	305.00		
Customer#000017746	17746.00	6882.00	1997-04-09
408446.93	303.00		
Customer#000013072	13072.00	1481925.00	1998-03-15
399195.47	301.00		
Customer#000082441	82441.00	857959.00	1994-02-07
382579.74	305.00		
Customer#000088703	88703.00	2995076.00	1994-01-30
363812.12	302.00		

57 rows processed.
Query Processed in 5.39 seconds.

Ended Executing this Stream at Thu Apr 11 11:03:07 2002

Stream Started at 1018540982.52
Stream Ended at 1018540987.91
Stream Processed in 5.39 seconds

SQL statements processed: 1

mqs00q19

Begin Execution at Thu Apr 11 11:03:08 2002

-- using default substitutions

```

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

```

REVENUE
3083843.06

1 row_ processed.
Query Processed in 2.43 seconds.

Ended Executing this Stream at Thu Apr 11 11:03:10 2002

Stream Started at 1018540988.07
Stream Ended at 1018540990.50
Stream Processed in 2.43 seconds

SQL statements processed: 1

mqs00q20

Begin Execution at Thu Apr 11 11:03:10 2002

-- using default substitutions

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

S_NAME      S_ADDRESS
Supplier#000000020 iybAE,RmTymrZVYaFZva2SH,j
Supplier#000000091 YV45D7TkfdQanOOZ7q9QxkyGUapU1oOWU6q3
Supplier#000000197 YC2Acon6kjY3zj3Fbxs2k4Vdf7X0cd2F
Supplier#000000226 83qOdU2EYRdPQAQhEtn GRZEd
Supplier#000000285 Br7e1nnt1yxrw6ImgpJ7YdhFDjuBf
Supplier#000000378 FfbhyCxWvcPrO8ltp9
Supplier#000000402 i9Sw4DoyMzhzKXCH9By,AYSgmD
Supplier#000000530 0qwCMwobKY OcmLyfRXlagA8ukENJv,
Supplier#000000688 D fw5ocppmZyYBBIPi718hCihLDZ5KhKX
Supplier#000000710 f19YPvOyb QoYwjKC,oPycpGfieBAcwKJo
Supplier#000000736 l6i2nMwVuoovKnuVgaSGK2rDy65DlAFLegiL7
Supplier#000000761 zLSLeIQUj2XrvTTFnnv7WAcYZGvvMTx882d4
Supplier#000000884 bhmEShejaS
Supplier#000000887 urEaTejH5POADP2ARrf
Supplier#000000935 ij98czM 2KzWe7dDT0xB8sq0UfCdvrX
Supplier#000000975 ,AC e,tBpNwKb5xMUzeohxlRn, hdZJo73gFQF8y
Supplier#000001263 rQWr6nf8ZhB2TAiIDlvo5lo
Supplier#000001399 LmrocnlMSyYOWuANx7
Supplier#000001446 lch9HMNU1R7a0LIybsUodVknk6
Supplier#000001454 TOPingu2TVXlJhl93h,
Supplier#000001500 wDmF5xLxtQch9ctVu,
Supplier#000001602 uKNWleafaM644
Supplier#000001626 UhxrNRzUu1dtFmp0
Supplier#000001682 pXTkGxrTQVvYH1Rr
Supplier#000001699 Q9C4rfJ26oijVPqqcqVXeRI
Supplier#000001700 7hMiCof1Y5zLFg
Supplier#000001726 TeRY7TtTH24sEword7yAaSkjx8
Supplier#000001730 Rc8e,1Pybn r6zo0VJIEiD0UD vhk
Supplier#000001746 qWsendlOekQG1aW4uq06uQaCm51se8lrv7 hBRd
Supplier#000001752 Fra7outx41THYJaRThdOGiBk
Supplier#000001856 jXcRgzYF0ah05iR8p6w5SbJLcUGyYiURPvFwUWM
Supplier#000001931 FpJbMU2h6ZR2eBv8I9NlxF
Supplier#000001939 Nrk,J4AbfReUs
*** Deleted Rows Here ***
```

```
Supplier#000006985 PrUUiBoQpy,OtgJ01Z4BxJQUyrw9c3I
Supplier#000007072 2tRyX9M1a 4Rcm57s779F1ANG9jlpK
Supplier#000007098 G3j8g0KC4OcbAu20VoPhrXQWMCUdjg8wgCHOExu
Supplier#000007135 ls DoKV7V5ulfQy9V
Supplier#000007160 TqDGBULB3cTqIT6FKDvm9BS4e4v,zwYiQPb
Supplier#000007169 tEc95D2moN9S84nd55O,dlnW
Supplier#000007322 wr7dgte5q MAjiY0uwm3MyDkSMX1
Supplier#000007365 51xhROLvQMj05DndtZWt
Supplier#000007398 V8eE6oZ00OFNU,
Supplier#000007402 4UVv58ery1rjmQSR5
Supplier#000007448 yhhpWiJi7EJ6Q5VCaQ
Supplier#000007477 9m9j0wfhWzCvVHxkU,PpAxwSH0h
Supplier#000007509 q8,V6LJR0HJjHCouSG7aLTMg
Supplier#000007561 rMcFg2530VC
Supplier#000007789 rQ7cUcPtudOyO3svNSkimqH6qrfWT2Sz
Supplier#000007801 69fi,U1r6nUb
Supplier#000007818 yhhc2CQec Jrvc8zqBi83
Supplier#000007885 u3sicch5ZpyTUpN1cJKNcAoabIWgY
Supplier#000007918 r,v9mBQ6LoEYyj1
Supplier#000007926 ErzCF80K9Uy
Supplier#000007957 ELwnio14ssoU1 dRyZIL OK3Vtzb
Supplier#000007965 F7Un5U7p5hhj
Supplier#000007968 DsF9UIZ2Fo6HXN9aErvg1ikHoD582HSGZpP
Supplier#000007998 LnASFBYRF0o9d6d,asBvVq9Lo2P
Supplier#000008168 aOa82a8ZbKcNfDLX
Supplier#000008231 IK7eGw Yj90sTdpSP,vcqWxLB
Supplier#000008243 2AyePMkDqmvZgJtZxtHfLo8h EiudCMxOmIIG
Supplier#000008275 BlbNDfWg,gpXKQILN
Supplier#000008323 75I18sZmASwm POeheRMdj9tmpyeQ,BfCXN5BIAb
Supplier#000008366 h778cEj14BuW9OEKlvPTWq4iwASR6EBBXN7zeS8
Supplier#000008423 RQhKnkAhR0DAR3lx4Q1weMMn00hNe Kq
Supplier#000008480 4sSDA4ACReklNJEm5T6b
Supplier#000008532 Uc29q4,5xVdDOF87UZrxhr4xWS0ihEUXuh
Supplier#000008595 MH0iB73GQ3z UW3O DbCbqmc
Supplier#000008610 SgVgP90vP452sUNTgzL9zKwXHXAzV6tV
Supplier#000008705 aE,trRNdPx,4yinTD9O3DebDlp
Supplier#000008742 HmPIQeZKCPECtUL14,kKq
Supplier#000008841 I 85Lu1sekbq2xrSlzm0
Supplier#000008895 2cH4okfaLSZTT7g8sKRbbJQxkmeFu2Esj
Supplier#000008967 2kwEHYMG 7FwozNImAUE6mH0hYtqYculJM
Supplier#000008972 w2vF6 D5YZO3visPXsqVfLADTK
Supplier#000009032 qK,trB6Sdy4Dz1BRUFNy
Supplier#000009147 rOAuryHxpZ9eOvx
Supplier#000009252 F7cZaPUHwh1 ZKzy3xmAVWC1XdP ue1p5m,i
Supplier#000009278 RqYTzgxj93CLX 0mcYfCENOfED
Supplier#000009327 uoqMdf7e7Gj9dbQ53
Supplier#000009430 igRqmneFt
Supplier#000009567 r4Wfx4c3xsEAjcgJ71HHZByornl D9vrztXlv4
Supplier#000009601 51m637bO,Rw5DnHWFUvLacRx9
Supplier#000009709 rRnCbHYgDgl9PZYnyWKVYSUW0vKg
Supplier#000009753 wLhVEcRmd7PkJF4FBnGK7Z
Supplier#000009796 z,y4ldmr15DOvPUqYG
Supplier#000009799 4wNjXGa4OKWl
Supplier#000009811 E3iuyq7UnZxU7oPZle2Gu6
Supplier#000009812 APFRMy3lCbgfGa53n5t9DxzFPQPgnjrGt32
Supplier#000009862 rJzweWeN58
Supplier#000009868 ROjGgx5gvtkmnUOueyy7v
Supplier#000009869 uclQxrpBTRMewGSM29t0rNTM30g1Tu3Xgg3mKag
Supplier#000009899 7XdpAHrzr1t,UQFEZ
Supplier#000009974 7wJ,J5DKcxSU4Kp1cQLpbCavB5AsvKT
```

204 rows processed.
Query Processed in 1.44 seconds.

Ended Executing this Stream at Thu Apr 11 11:03:12 2002

Stream Started at 1018540990.65
Stream Ended at 1018540992.09
Stream Processed in 1.44 seconds

SQL statements processed: 1

mqs00q21

Begin Execution at Thu Apr 11 11:03:12 2002

-- using default substitutions

```
select * from (
select
s_name,
count(*) numwait
from
supplier,
lineitem l1,
orders,
nation
where
s_suppkey = l1.l_suppkey
and o_orderkey = l1.l_orderkey
and o_orderstatus = 'F'
and l1_receiptdate > l1.l_commitdate
and exists (
select
*
from
lineitem l2
where
l2.l_orderkey = l1.l_orderkey
and l2.l_suppkey <> l1.l_suppkey
)
and not exists (
select
*
from
lineitem l3
where
l3.l_orderkey = l1.l_orderkey
and l3.l_suppkey <> l1.l_suppkey
and l3.l_receiptdate > l3.l_commitdate
)
and s_nationkey = n_nationkey
and n_name = 'SAUDI ARABIA'
group by
s_name
order by
numwait desc,
s_name)
where rownum <= 100
```

S_NAME	NUMWAIT
Supplier#000002829	20.00
Supplier#000005808	18.00
Supplier#000000262	17.00
Supplier#000000496	17.00
Supplier#000002160	17.00
Supplier#000002301	17.00
Supplier#000002540	17.00
Supplier#000003063	17.00
Supplier#000005178	17.00
Supplier#000008331	17.00
Supplier#000002005	16.00
Supplier#000002095	16.00
Supplier#000005799	16.00
Supplier#000005842	16.00
Supplier#000006450	16.00
Supplier#000006939	16.00
Supplier#000009200	16.00
Supplier#000009727	16.00
Supplier#000000486	15.00
Supplier#000000565	15.00
Supplier#000001046	15.00
Supplier#000001047	15.00
Supplier#000001161	15.00
Supplier#000001336	15.00
Supplier#000001435	15.00
Supplier#000003075	15.00
Supplier#000003335	15.00
Supplier#000005649	15.00
Supplier#000006027	15.00

Supplier#000006795	15.00
Supplier#000006800	15.00
Supplier#000006824	15.00
Supplier#000007131	15.00
Supplier#000007382	15.00
Supplier#000008913	15.00
Supplier#000009787	15.00
Supplier#000006633	14.00
Supplier#000001960	14.00
Supplier#000002323	14.00
Supplier#000002490	14.00
Supplier#000002993	14.00
Supplier#000003101	14.00
Supplier#000004489	14.00
Supplier#000005435	14.00
Supplier#000005583	14.00
Supplier#000005774	14.00
Supplier#000007579	14.00
Supplier#000008180	14.00
Supplier#000008695	14.00
Supplier#000009224	14.00
Supplier#000000357	13.00
Supplier#000000436	13.00
Supplier#000000610	13.00
Supplier#000000788	13.00
Supplier#000000889	13.00
Supplier#000001062	13.00
Supplier#000001498	13.00
Supplier#000002056	13.00
Supplier#000002312	13.00
Supplier#000002344	13.00
Supplier#000002596	13.00
Supplier#000002615	13.00
Supplier#000002978	13.00
Supplier#000003048	13.00
Supplier#000003234	13.00
Supplier#000003727	13.00
Supplier#000003806	13.00
Supplier#000004472	13.00
Supplier#000005236	13.00
Supplier#000005906	13.00
Supplier#000006241	13.00
Supplier#000006326	13.00
Supplier#000006384	13.00
Supplier#000006394	13.00
Supplier#000006624	13.00
Supplier#000006629	13.00
Supplier#000006682	13.00
Supplier#000006737	13.00
Supplier#000006825	13.00
Supplier#000007021	13.00
Supplier#000007417	13.00
Supplier#000007497	13.00
Supplier#000007602	13.00
Supplier#000008134	13.00
Supplier#000008234	13.00
Supplier#000009435	13.00
Supplier#000009436	13.00
Supplier#000009564	13.00
Supplier#000009896	13.00
Supplier#000000379	12.00
Supplier#000000673	12.00
Supplier#000000762	12.00
Supplier#000000811	12.00
Supplier#000000821	12.00
Supplier#000001337	12.00
Supplier#000001916	12.00
Supplier#000001925	12.00
Supplier#000002039	12.00
Supplier#000002357	12.00
Supplier#000002483	12.00

100 rows processed.

Query Processed in 6.75 seconds.

Ended Executing this Stream at Thu Apr 11 11:03:19 2002

Stream Started at 1018540992.30
Stream Ended at 1018540999.05
Stream Processed in 6.75 seconds

SQL statements processed: 1

mqs00q22

Begin Execution at Thu Apr 11 11:03:19 2002

-- using default substitutions

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

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

7 rows processed.
Query Processed in 2.11 seconds.

Ended Executing this Stream at Thu Apr 11 11:03:21 2002

Stream Started at 1018540999.20
Stream Ended at 1018541001.31
Stream Processed in 2.11 seconds

SQL statements processed: 1

Appendix D: Seed Values and Query Substitution Parameters

seed_values

Stream 00: 624180406
Stream 01: 624180407
Stream 02: 624180408
Stream 03: 624180409
Stream 04: 624180410
Stream 05: 624180411

m2param.0

14	1993-06-01					
2	40	STEEL	ASIA			
9	drab					
20	asure	1995-01-01	ARGENTINA			
6	1996-01-01	0.06	24			
17	Brand#53	WRAP CAN				
18	314					
8	INDONESIA		ASIA	SMALL POLISHED NICKEL		
21	IRAN					
13	express	packages				
3	FURNITURE	1995-03-18				
22	10	11	16	22	31	25
	21					
16	Brand#15	LARGE BURNISHED	47	27	14	
	3	1	13	16	17	
4	1995-09-01					
11	ALGERIA	0.0000010000				
15	1996-08-01					
1	60					
10	1993-12-01					
19	Brand#51	Brand#31	Brand#12	4	19	24
5	MIDDLE EAST	1996-01-01				
7	KENYA	INDONESIA				
12	REG AIR	SHIP	1993-01-01			

m2param.1

21	BRAZIL					
3	AUTOMOBILE	1995-03-03				
18	315					
5	AFRICA	1996-01-01				
11	JAPAN	0.0000010000				
7	FRANCE	ARGENTINA				
6	1996-01-01	0.04	24			
20	lavender	1993-01-01	MOZAMBIQUE			
17	Brand#15	SM CASE				
12	SHIP	REG AIR	1993-01-01			
16	Brand#41	PROMO POLISHED	35	8	45	
	44	48	2	36	37	
15	1994-05-01					
13	special	packages				
10	1994-09-01					
2	27	BRASS	AFRICA			
8	ARGENTINA		AMERICA	SMALL BURNISHED NICKEL		
14	1993-09-01					
19	Brand#13	Brand#14	Brand#51	9	20	20
9	cream					
22	14	30	12	32	27	26
	13					
1	68					
4	1993-06-01					

m2param.2

6	1996-01-01	0.09	25			
17	Brand#12	SM JAR				
14	1993-12-01					
16	Brand#21	SMALL BRUSHED	4	45	20	
	22	44	5	19	43	
19	Brand#15	Brand#52	Brand#55	4	10	27
10	1993-06-01					
9	chartreuse					
2	15	TIN	EUROPE			
15	1996-12-01					
8	CHINA	ASIA	STANDARD BRUSHED NICKEL			
5	AMERICA	1996-01-01				
22	14	34	33	19	21	27
	17					
12	FOB	REG AIR	1993-01-01			
7	UNITED KINGDOM	CHINA				
13	special	packages				
18	313					
1	76					
4	1996-01-01					
20	smoke	1997-01-01	FRANCE			
3	FURNITURE	1995-03-20				
11	ALGERIA	0.0000010000				
21	SAUDI ARABIA					

m1param.3

8	IRAN	MIDDLE EAST	STANDARD PLATED NICKEL			
5	ASIA	1996-01-01				
4	1993-10-01					
6	1996-01-01	0.07	24			
17	Brand#14	SM CAN				
7	MOROCCO	IRAN				
1	84					
18	314					
22	17	21	29	20	32	31
	27					
14	1994-03-01					
9	blanched					
10	1994-03-01					
15	1994-08-01					
11	JORDAN	0.0000010000				
20	firebrick	1995-01-01	SAUDI ARABIA			
2	3	COPPER	AFRICA			
21	JAPAN					
19	Brand#12	Brand#35	Brand#54	10	11	23
13	special	packages				
16	Brand#11	ECONOMY BURNISHED				
	8	41	15	37	0	23
12	MAIL	REG AIR	1994-01-01			
3	MACHINERY	1995-03-05				

m2param.4

5	EUROPE	1997-01-01				
21	EGYPT					
14	1994-06-01					
19	Brand#24	Brand#13	Brand#44	5	12	30
15	1997-03-01					
17	Brand#11	LG CASE				
12	TRUCK	REG AIR	1994-01-01			
6	1997-01-01	0.04	24			
4	1996-05-01					
9	antique					
8	BRAZIL	AMERICA	STANDARD ANODIZED NICKEL			
16	Brand#41	STANDARD PLATED	6	35	8	
	20	41	30	2	0	
11	ARGENTINA	0.0000010000				
2	41	STEEL	EUROPE			
10	1995-01-01					
18	312					
1	92					

13	special requests					
7	GERMANY		BRAZIL			
22	20	28	17	10	13	23
	30					
3	BUILDING	1995-03-22				
20	pink	1993-01-01	IRAN			

m2param.5

21	VIETNAM					
15	1994-12-01					
4	1995-02-01					
6	1997-01-01	0.09	25			
7	UNITED STATES		ROMANIA			
16	Brand#21	MEDIUM BRUSHED	18	6		43
	45	37	11	29	10	
19	Brand#21	Brand#51	Brand#43	10	13	27
18	313					
14	1994-10-01					
22	25	27	18	28	16	26
	12					
11	KENYA	0.0000010000				
13	special requests					
3	MACHINERY		1995-03-07			
1	100					
2	29	BRASS	AMERICA			
5	MIDDLE EAST		1997-01-01			
8	ROMANIA EUROPE		PROMO POLISHED BRASS			
20	burlywood	1997-01-01	ALGERIA			
12	AIR	RAIL	1994-01-01			
17	Brand#13	LG JAR				
10	1993-10-01					
9	turquoise					

Appendix E: Implementation-Specific Layer/Driver Code

runTPCHall

```
#!/bin/ksh

ECHO=echo

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

RUN_ID_FILE=${KIT_DIR}/audit/r_id

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

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

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

SCRIPT_LOG_FILE=${OUT_DIR}/main.out
RDB_TABLES=${OUT_DIR}/rdbtablest.log
FIRST_TEN=${OUT_DIR}/firstten.log
CHECK_IDX=${OUT_DIR}/checkidx.log

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_TPC1.log"
>> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE
mv $ORACLE_HOME/rdbms/log/alert_TPC1.log
$ORACLE_HOME/rdbms/log/alert_TPC1.log.preAudit.$RUN_ID
touch $ORACLE_HOME/rdbms/log/alert_TPC1.log

echo >> $SCRIPT_LOG_FILE
echo "Starting a new Oracle log file: $ORACLE_HOME/rdbms/log/alert_TPC2.log"
>> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE
mv $ORACLE_HOME/rdbms/log/alert_TPC2.log
$ORACLE_HOME/rdbms/log/alert_TPC2.log.preAudit.$RUN_ID
touch $ORACLE_HOME/rdbms/log/alert_TPC2.log

echo >> $SCRIPT_LOG_FILE
echo "Starting a new Oracle log file: $ORACLE_HOME/rdbms/log/alert_TPC3.log"
>> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE
mv $ORACLE_HOME/rdbms/log/alert_TPC3.log
$ORACLE_HOME/rdbms/log/alert_TPC3.log.preAudit.$RUN_ID
touch $ORACLE_HOME/rdbms/log/alert_TPC3.log

echo >> $SCRIPT_LOG_FILE
echo "Starting a new Oracle log file: $ORACLE_HOME/rdbms/log/alert_TPC4.log"
>> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE
mv $ORACLE_HOME/rdbms/log/alert_TPC4.log
$ORACLE_HOME/rdbms/log/alert_TPC4.log.preAudit.$RUN_ID
touch $ORACLE_HOME/rdbms/log/alert_TPC4.log

echo "Start: load database `date`" >> $SCRIPT_LOG_FILE
```

```
$SF/schema_load/create_database_4node.sh >
${OUT_DIR}/create_database_4node.log

$SF/scripts/shutdown_instance.sh
$SF/scripts/startup_4_instances.sh

STIME=`$GTIME`
echo "Start: timed load portion `date`" >> $SCRIPT_LOG_FILE

$SF/schema_load/database_load_tables_et.sh >
${OUT_DIR}/database_load_tables_et.log
$SF/schema_load/create_indexes_all.sh > ${OUT_DIR}/create_indexes_all.log
$SF/schema_load/analyze_ops.sh > ${OUT_DIR}/analyze_ops.log

echo "End: timed load portion `date`" >> $SCRIPT_LOG_FILE

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

echo "Start: dbtables.sql, firstten.sql, checkidx.sql (BEFORE) `date`" >>
$SCRIPT_LOG_FILE
sqlplus ${DATABASE_USER} @$KIT_DIR/audit/dbtables >
${RDB_TABLES}.before_run 2>&1
sqlplus ${DATABASE_USER} @$KIT_DIR/audit/firstten >
${FIRST_TEN}.before_run 2>&1
sqlplus ${DATABASE_USER} @$KIT_DIR/audit/checkidx >
${CHECK_IDX}.before_run 2>&1
echo "End: dbtables.sql, firstten.sql, checkidx.sql (BEFORE) `date`" >>
$SCRIPT_LOG_FILE

runTPCHpt ${SCALE_FACTOR} 1 ${RUN_ID}

runTPCHpt ${SCALE_FACTOR} 2 ${RUN_ID}

echo "Start: dbtables.sql and checkidx.sql (AFTER) `date`" >> $SCRIPT_LOG_FILE
sqlplus ${DATABASE_USER} @$KIT_DIR/audit/dbtables >
${RDB_TABLES}.after_run 2>&1
sqlplus ${DATABASE_USER} @$KIT_DIR/audit/checkidx >
${CHECK_IDX}.after_run 2>&1
echo "End: dbtables.sql and checkidx.sql (AFTER) `date`" >> $SCRIPT_LOG_FILE

${SF}/scripts/kill.sh sct

cp $ORACLE_HOME/rdbms/log/alert_TPC1.log $OUT_DIR
cp $ORACLE_HOME/rdbms/log/alert_TPC2.log $OUT_DIR
cp $ORACLE_HOME/rdbms/log/alert_TPC3.log $OUT_DIR
cp $ORACLE_HOME/rdbms/log/alert_TPC4.log $OUT_DIR

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

runTPCHpt

#!/bin/ksh
#set -x
#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

TPCH_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}
#OUT_DIR=/flatfiles/results/${RUN_ID}
if [ ! -d $OUT_DIR ]
then
mkdir $OUT_DIR
fi

TPCH_LOG=${OUT_DIR}
TPCH_RPT=${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"

TPCH_LOG_FILE=${TPCH_LOG}/m${PARA}s0
TPCH_RPT_FILE=${TPCH_RPT}/m${PARA}s0inter
QRY_FILE=${TPCH_RPT}/qtemp.${PARA}s0
QUERY_PARAMETER=${TPCH_LOG}/qp${PARA}.0
SCRIPT_LOG_FILE=${TPCH_LOG}/m${PARA}timing

```

```

UF1_LOG=${TPCH_LOG}/m${PARA}s0rf1
UF2_LOG=${TPCH_LOG}/m${PARA}s0rf2
STREAM_COUNT_LOG=${TPCH_LOG}/m${PARA}tstrcnt

echo "TPC-H Test - RUN:${PARA} SEQUENCE:${RUN_ID} `date`" >
$SCRIPT_LOG_FILE
echo "TPC-H Test - RUN:${PARA} SEQUENCE:${RUN_ID} `date`" >
$TPCH_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 >> ${TPCH_RPT_FILE}
echo Start Time: $UF1_START, $SDATE >> ${TPCH_RPT_FILE}
echo End Time: $UF1_END, $E1DATE >> ${TPCH_RPT_FILE}
echo "" >> ${TPCH_RPT_FILE}

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

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

${QPROG} ${DATABASE_USER} q${QRY_FILE} l${TPCH_LOG_FILE}
r${TPCH_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 $UF1_START, `date`" >> $SCRIPT_LOG_FILE
${ECHO} ${UPD_SPT}/runuf2.sh ${UF_SET} >> $UF2_LOG 2>&1
UF2_END=`$GTIME`
END=`$GTIME`
EDATE=`date`

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

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

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

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

i=$START_SET
PSEED=`cat $SEED_FILE`

```

```

while [ $i -le $STOP_SET ]; do
    TPCD_LOG_FILE=${TPCH_LOG}/mt${RUN_ID}_${i}.log
    TPCD_RPT_FILE=${TPCH_RPT}/mt${RUN_ID}_${i}.rpt
    QUERY_PARAMETER=${TPCH_LOG}/qp${PARAM}.s${i}
    QRY_FILE=${TPCH_RPT}/qtemp.${PARAM}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:${PARAM} SEQUENCE:${RUN_ID}
$TH_START_T, $TH_START_D" >> $SCRIPT_LOG_FILE

# starts a script to count the streams during the throughput run
(scnt.sh $PARAM $RUN_ID > $STREAM_COUNT_LOG &)

while [ $i -le $STOP_SET ]; do
    M_SDATE=`date`
    M_STIME=`$(GTIME)`
    TPCD_LOG_FILE=${TPCH_LOG}/m${PARAM}s${i}
    TPCD_RPT_FILE=${TPCH_RPT}/m${PARAM}s${i}
    echo "Start Query Stream $i M_STIME, ${M_SDATE}" >> $SCRIPT_LOG_FILE
    QRY_FILE=${TPCH_RPT}/qtemp.${PARAM}s${i}
    ${QPROG} ${DATABASE_USER} q${QRY_FILE} l${TPCH_LOG_FILE}
    r${TPCH_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} $PARAM >> $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 $STOP_SET ]; do
    TPCD_LOG_FILE=${TPCH_LOG}/m${PARAM}s${i}
    ${KIT_DIR}/audit/abridge.pl ${TPCH_LOG_FILE}
    i=`expr $i + 1`
done
#kill -9 `ps -ef | grep scnt.sh | grep -v grep | awk '{print $2}`
#${SF}/scripts/kill.sh scnt.sh
#calculate the metric
#analyze_streams.pl -f p -n $RUN_ID >
${TPCH_RPT}/tpch_metric.${RUN_ID}.${HID}.rpt

```

runTPCHus

```
#!/bin/ksh
```

```

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

DSS_QUERY=${KIT_DIR}/queries
export DSS_QUERY

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

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

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

GTIME=${SRC_DIR}/gtime
HID=1

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

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

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

    # Execute UF1

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

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

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

    # Execute UF2

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

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

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

```

done

print > /tmp/th_pipe2

runuf1.sh

```
#!/bin/ksh
#
# $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.>
#
#
. $KIT_DIR/env
O=${ORACLE_HOME}
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
LOG_DIR=${UPDATE_DIR}/log
GTIME=gtime
#SF=${SCALE_FACTOR}
#PAR_HINT=${UPDATE_DOP}
PAR_HINT=16

LOGPATH=.
#PASSWD=${DATABASE_USER}
PASSWD="tpcd/tpcd"

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

# perform the update function 1

START=`$GTIME`
echo "Update Function 1 Set $SETNUM Begin Time = " `date`

# first create the temp tables

sqlplus /NOLOG << !

connect $PASSWD;
set timing on
set serveroutput on
set echo on
drop directory data_dir;
create directory data_dir as /flatfiles/update_sets';

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

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

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

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

alter session force parallel dml parallel (degree ${PAR_HINT});
alter session set isolation_level = serializable;
alter session set optimizer_index_cost_adj = 25;

commit;

insert into orders (select * from temp_o_et);

insert into lineitem (select * from temp_l_et);

commit;

drop table temp_l_et;
drop table temp_o_et;

exit;
!

END=`$GTIME`

# Done

echo ""
echo "Update Function 1 Set $SETNUM End Time = " `date`
```

```
echo "Elapsed Time is `echo $END - $START | bc`"
echo ""
```

runuf2.sh

```
#!/bin/ksh
#
# $Header: runuf2.sh 25-oct-2001.15:56:05 mpoess Exp $
#
# runuf2.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation. All rights reserved.
#
# NAME
#   runuf2.sh - <one-line expansion of the name>
#
# DESCRIPTION
#   runuf2.sh [-u <uid/passwd to login>] [-p <program>] <run_id>
#               <scale factor> <pair number> <parallelism>
# USAGE
#   To execute UF2.
#
# NOTES
#   <other useful comments, qualifications, etc.>
#
. $KIT_DIR/env
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
GTIME=gtime
LOG_DIR=${UPDATE_DIR}/log
PAR_HINT=16
PASSWD="tpcd/tpcd"

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

SETNUM=$1

i=1
PID=""

START=`$GTIME`
echo "Update Function 2 Set $SETNUM Begin Time = " `date`
# first create the temp tables

sqlplus /NOLOG << !

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

drop directory data_dir;
create directory data_dir as '/flatfiles/update_sets';

drop table temp_okey_et;
drop table temp_okey;

create table temp_okey_et(
    t_orderkey      number
)
organization external (
    type ORACLE_LOADER
    default directory data_dir
    access parameters
    (
        records delimited by newline
        badfile 'okey.${SETNUM}.bad'
        logfile 'okey.${SETNUM}.log'
        fields terminated by '|'
        missing field values are null
    )
)
location (
    'delete.${SETNUM}.log'
)
reject limit unlimited;
```

```
alter table temp_okey_et parallel ${PAR_HINT};
```

```
create table temp_okey parallel ${PAR_HINT} nologging as select * from
temp_okey_et;
```

```
create unique index i_temp_okey on temp_okey (t_orderkey) parallel ${PAR_HINT}
nologging compute statistics;
alter index i_temp_okey parallel;
analyze table temp_okey estimate statistics sample 2 percent;
alter session force_parallel_dml parallel ${PAR_HINT};
alter session set isolation_level = serializable;
alter session set optimizer_index_cost_adj = 25;
```

```
delete from (select /*+ ordered index(o) use_nl(o) */ o.rowid from orders o,
temp_okey t where o.o_orderkey = t.t_orderkey order by 1);
```

```
delete from (select /*+ ordered index(l) use_nl(l) */ l.rowid from lineitem l,temp_okey
t where l.l_orderkey = t.t_orderkey order by 1);
```

```
commit;
```

```
drop table temp_okey;
drop table temp_okey_et;
exit;
!
```

```
END=`$GTIME`
```

```
# Done
```

```
echo ""
echo "Update Function 2 Set $SETNUM End Time = " `date`
echo "Elapsed Time is `echo $END - $START | bc`"
echo ""
```

shutdown_instance.sh

```
#!/bin/ksh

sqlplus <<EOF
/ as sysdba
shutdown
exit
EOF
```

startup_4_instances.sh

```
#!/bin/ksh

echo "Starting Up Oracle on tpchorc1 Now..."

$SF/scripts/startup_tpchorc1.sh

echo "...Oracle Startup Complete on tpchorc1"

sleep 3

echo "Starting Up Oracle on tpchorc2 Now..."

rsh tpchorc2 $SF/scripts/startup_tpchorc2.sh

echo "...Oracle Startup Complete on tpchorc2"

sleep 3

echo "Starting Up Oracle on tpchorc3 Now..."

rsh tpchorc3 $SF/scripts/startup_tpchorc3.sh

echo "...Oracle Startup Complete on tpchorc3"

sleep 3

echo "Starting Up Oracle on tpchorc4 Now..."

rsh tpchorc4 $SF/scripts/startup_tpchorc4.sh
```

```
echo "...Oracle Startup Complete on tpchorc4"
```

startup_tpchorc1.sh

```
#!/bin/ksh

sqlplus <<EOF
/ as sysdba
startup parallel pfile=$SF/control_files/tpchorc1.ora
exit
EOF
```

startup_tpchorc2.sh

```
#!/bin/ksh

sqlplus <<EOF
/ as sysdba
startup parallel pfile=$SF/control_files/tpchorc2.ora
exit
EOF
```

startup_tpchorc3.sh

```
#!/bin/ksh

sqlplus <<EOF
/ as sysdba
startup parallel pfile=$SF/control_files/tpchorc3.ora
exit
EOF
```

startup_tpchorc4.sh

```
#!/bin/ksh

sqlplus <<EOF
/ as sysdba
startup parallel pfile=$SF/control_files/tpchorc4.ora
exit
EOF
```

env

EMPTY - All Oracle Environment Variables set at oracle user login

gtime.c

```
/* Copyright (c) Oracle Corporation 2001. 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 08/29/01 - Creation
elspeed 01/16/02 - Modify for Compaq Tru64

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

# include <sys/time.h>

main ()
{

struct timeval tp;

    gettimeofday(&tp, NULL);

    printf ("%lld\n", tp.tv_sec) ;
/*
    printf ("usec = %ld\n", tp.tv_usec) ;
*/
}

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

qexecpl.c

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

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

/*

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

DESCRIPTION
    SQL Execution Engine, Oracle v8, OCI version

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

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

```
*/

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

#include "qexecpl.h"

/* Function Prototypes */

extern double gtime();
```



```

/* function prototypes from gen.c */

int get_statement();

/* Declare error handling functions */

void sql_error();

/* Other prototypes */

int define_output_variables();
void process_select_list();
void usage();
void SQLInit();
void SQLExec();
void SQLExit();
void *memalloc();
void print_header();
void print_rows();
int OFEN();
void remove_newline();

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

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

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

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

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

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

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

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

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

char stmt[SQL_LEN]; /* The SQL statement or comment line. */
char qn[3]; /* Number of the query being executed */
char qnp[3]; /* Number of the previous query executed */
char cmt[5000]; /* Buffer to save the comment. */
#ifdef LINUX
FILE *qtemp; /* fd for query template */
FILE *logfile; /* log and report files */
FILE *rep;
#else
FILE *qtemp = stdin; /* fd for query template */
FILE *logfile = stdout; /* log and report files */
FILE *rep = stdout;
#endif
void *defbuf; /* Buffer pointer for ODEFIN */
int deflen = 0; /* Size of data type for ODEFIN */
int deftype = 1; /* Oracle type number for ODEFIN */

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

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

/* OCI handles */

OCISvcCtx *tpcenv = NULL;
OCIServer *tpcsrv = NULL;
OCIError *errhp = NULL;
OCISvcCtx *tpcsvc = NULL;
OCISession *tpcusr = NULL;

```

```

OCISmt *curq = NULL;
OCISmt *cur_dml = NULL;
OCISmt *cur_ddl = NULL;
OCIParam *tpcparr = 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) OCIErrGet(errhp, 1, NULL, (sb4*)&errcode, (text*)msg,
                                2048, OCI_HTYPE_ERROR);
            else
                (void) OCIErrGet(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) OCIErrGet(errhp, 1, NULL, (sb4*)&errcode, (text*)msg,
                                2048, OCI_HTYPE_ERROR);
            else
                (void) OCIErrGet(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) OCIErrGet(errhp, 1, NULL, (sb4*)&errcode, (text*)msg,
                                2048, OCI_HTYPE_ERROR);
            else
                (void) OCIErrGet(errhp, 1, NULL, (sb4*)&errcode, (text*)msg,
                                2048, OCI_HTYPE_ENV);
            fprintf(stderr, "%s\n", msg);
            break;
    }

    /* Rollback just in case */

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

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

```

```

SQLexit();

exit(1);
}

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

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

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

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

    strcpy(logname, argv[1]);

    /* Process optional parameters */

    argc -= 1;
    argv += 1;

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

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

    SQLinit();

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

    time(&tim);
    fprintf(logfile, "Begin Execution at %s\n", 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++;
        fflush(rep);
        fflush(logfile);
        /*
        fprintf(logfile,"Statement Started at %.2f\n", s_tr_start);
        fprintf(logfile,"Statement Ended at %.2f\n", s_tr_end);

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

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

/* Get Timing for the last query */

tr_end = gettimeofday();

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\n", (tr_end - tr_start));

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

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

fflush(rep);
fflush(logfile);

/* Close the query template file */

fclose(qtemp);

/* Disconnect from ORACLE. */

SQLexit();
exit(0);
}

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

void SQLinit() {

    int i;

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

    for (i=0; i<MAX_SEL_LIST; i++) {
        if (i < MAX_PREALLOC) {
            dlist[i] = (dtype *) memalloc (sizeof(dtype));
            dlist[i]->defhdl = NULL;
        }
        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=OCIEEnvInit((OCIEEnv **) &tpcenv, OCI_DEFAULT, 0, (dvoid **)0)) !=
        OCI_SUCCESS)
        sql_error(tpcenv, status, 0);

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

    /* get username and password */

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

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

```

```

OCIaset(tpcsvc, OCI_HTYPE_SVCCTX, tpcsrv, 0, OCI_ATTR_SERVER, errhp);

OCIaset(tpcusr, OCI_HTYPE_SESSION, logname, strlen(logname), OCI_ATTR_USE
        RNAME,
        errhp);

OCIaset(tpcusr, OCI_HTYPE_SESSION, passwd, strlen(passwd), OCI_ATTR_PASSW
        ORD,
        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((OCIEEnv *)tpcenv, (OCIError *)errhp, (OCISvcCtx *)tpcsvc,
        (text *)logname, strlen(logname), (text *)passwd,
        strlen(passwd), (text *)0, 0)) != OCI_SUCCESS)
        sql_error(errhp, status, 1);
*/
    printf("\nConnected to ORACLE as user: %s\n\n", logname);
}

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

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

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

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

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

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

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

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

        fflush(logfile);
        fflush(rep);
    }
    else
        tr_start = 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 */

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

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

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

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

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

OCIsexec(tpcscv,curq,errhp,0);

num_sel_list = define_output_variables();

/* Executes the query and fetches the rows */

(void) process_select_list(num_sel_list);

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

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

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

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

/* reset set_fetchrows */

num_to_fetch = -1;
}

void SQLexit() {

    int i;

    OCILogoff(tpcscv,errhp);
    OCIhfree(tpcenv,OCI_HTYPE_STMT);
    OCIhfree(tpcscv,OCI_HTYPE_SVCCCTX);
    OCIhfree(tpcscv,OCI_HTYPE_SERVER);
    OCIhfree(tpcusr,OCI_HTYPE_SESSION);

    /* free all memory */

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

    /* Flush all output */

    fflush(rep);
    fflush(logfile);
}

}

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

int define_output_variables()
{
    int i;
    int retflag = 0;

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

        slist[i].buflen = MAX_COLNAME_SIZE;

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

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

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

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

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

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

        if (i >= MAX_PREALLOC) {
            dlist[i] = (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,er
rhp);

    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,
MAX_ARRAY:ntf),
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;
            }
        }
    } else {
        OCISmtFetch(curq, errhp, ntf, OCI_FETCH_NEXT, OCI_DEFAULT);
        OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,OCI_ATTR_ROW_COUNT,er
rhp);
        print_rows(num,rows_ret);
    }

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

int get_statement()
{
    char line[128];
    char *pos, *str;

/* Reset statement buffer */

stmt[0] = '\0';

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

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

/* remove blanks */

str = line;

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

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

strcat(stmt, str);

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

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

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

```

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

```
void *memalloc(size)
int size;
{
    void *tmp;

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

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

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

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

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

```
        /* extract the column name */
```

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

```
        /* format the output a little */
```

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

```
        /* do a little bit of formatting */
```

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

```
#ifdef FORMAT1
    if ((slist[i].dbtype == INT_TYPE) || (slist[i].dbtype == FLT_TYPE))
        fprintf(logfile, "%s ", cwid, slist[i].buf);
    else /* string type */
        fprintf(logfile, "%s ", -cwid, slist[i].buf);
#else
    fprintf(logfile, "%s ", -cwid, colname);
#endif /* FORMAT1 */
}
```

```
    fprintf(logfile, "\n");
}
```

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

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

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

```
        len = 0;
```

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

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

```
            /* do a little bit of formatting */
```

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

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

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

```
void remove_newline(str)
char *str;
{
    char *p;

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

qexecpl.h

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

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

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

```
/*
 * NAME
 * qexecpl.h
```

```
DESCRIPTION
SQL statement execution front-end header file.
```

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

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

```
EXAMPLES
```

```
NOTES
```

```

<other useful comments, qualifications, etc.>

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

*/

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

# define QSTREAMPL_H

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

# include <oratypes.h>

# include <oratypes.h>

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

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

/* some basic definitions */

# define UNAME_LEN 64
# define MAX_FILE_PATH_LEN 128

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

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

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

/* defines and typedefs for query description */

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

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

/* select list. */

/* types for describe */

# define CHAR_TYPE 1
# define NUM_TYPE 2
# define INT_TYPE 3
# define FLT_TYPE 4
# define STR_TYPE 5
# define DATE_TYPE 12

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

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

typedef struct des
{
    ub2 dbsize;
    ub4 buflen;
    /* sb2 dsize; */
    sb4 scale;
    /* sb2 nullok; */
    OCITextCode 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 OCIlaget(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 OCIsaset(hndl,htyp,attp,size,atyp,errh) \
    if((status=OCIAttrSet((dvoid *)hndl,htyp,(dvoid *)attp,size,atyp,errh)) != \
OCI_SUCCESS) \
        sql_error(errh,status,1); \
    else \
        DISCARD 0

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

#define ISOTXT "alter session set isolation_level = serializable"
#define PDMLTXT "alter session force parallel dml parallel (degree 84)"
#define PDDLTX "alter session force parallel ddl parallel (degree 84)"

#endif /* QSTREAMPL_H */

```


Appendix F

Activity occurring between the Database Load and the commencement of Run1...consists of SQL scripts executed primarily for the auditor.

checkidx.sql

```
set echo on
spool checkidx.out
select index_name from user_indexes;
spool off
exit;
```

dbtables.sql

```
set echo on
set numwidth 25
spool rdbtablest
SELECT COUNT(*) FROM LINEITEM;

SELECT * FROM LINEITEM
WHERE L_ORDERKEY IN
( 4, 26598, 148577, 387431, 56704, 517442, 600000)
AND L_LINENUMBER = 1
ORDER BY L_ORDERKEY;

SELECT * FROM REGION;

SELECT COUNT(*) FROM NATION;

SELECT * FROM NATION
WHERE N_NATIONKEY IN ( 3,10,14,20)
ORDER BY N_NATIONKEY;

SELECT COUNT(*) FROM ORDERS;

SELECT * FROM ORDERS
WHERE O_ORDERKEY IN ( 7, 44065, 287590, 411111, 483876, 599942 )
ORDER BY O_ORDERKEY;

SELECT COUNT(*) FROM PART;

SELECT * FROM PART
WHERE P_PARTKEY IN (1,984,8743,9028,13876,17899,20000)
ORDER BY P_PARTKEY;

SELECT COUNT(*) FROM PARTSUPP;

SELECT* FROM PARTSUPP
WHERE PS_PARTKEY = 3398
AND PS_SUPPKEY = (SELECT MIN(PS_SUPPKEY)
FROM PARTSUPP WHERE PS_PARTKEY = 3398);

SELECT* FROM PARTSUPP
WHERE PS_PARTKEY =15873
AND PS_SUPPKEY = (SELECT MIN(PS_SUPPKEY)
FROM PARTSUPP WHERE PS_PARTKEY = 15873);

SELECT* FROM PARTSUPP
```

```
WHERE PS_PARTKEY = 11394
AND PS_SUPPKEY = (SELECT MIN(PS_SUPPKEY)
FROM PARTSUPP WHERE PS_PARTKEY = 11394);

SELECT* FROM PARTSUPP
WHERE PS_PARTKEY = 6743
AND PS_SUPPKEY = (SELECT MIN(PS_SUPPKEY)
FROM PARTSUPP WHERE PS_PARTKEY = 6743);

SELECT* FROM PARTSUPP
WHERE PS_PARTKEY = 19763
AND PS_SUPPKEY = (SELECT MIN(PS_SUPPKEY)
FROM PARTSUPP WHERE PS_PARTKEY =19763);
```

```
SELECT COUNT(*) FROM SUPPLIER;
```

```
SELECT * FROM SUPPLIER
WHERE S_SUPPKEY IN (83,265,492,784,901,1000)
ORDER BY S_SUPPKEY;
```

```
DROP TABLE MINMAX;
```

```
CREATE TABLE MINMAX
(TNAME CHAR(15),
KEYMIN INTEGER,
KEYMAX INTEGER);
```

```
INSERT INTO MINMAX
SELECT 'LINEITEM_ORD',MIN(L_ORDERKEY),MAX(L_ORDERKEY)
FROM LINEITEM ;
```

```
INSERT INTO MINMAX
SELECT 'LINEITEM_NBR',MIN(L_LINENUMBER),MAX(L_LINENUMBER)
FROM LINEITEM;
```

```
INSERT INTO MINMAX
SELECT 'ORDERTBL',MIN(O_ORDERKEY),MAX(O_ORDERKEY)
FROM ORDERS;
```

```
INSERT INTO MINMAX
SELECT 'CUSTOMER',MIN(C_CUSTKEY),MAX(C_CUSTKEY)
FROM CUSTOMER;
```

```
INSERT INTO MINMAX
SELECT 'PART',MIN(P_PARTKEY),MAX(P_PARTKEY)
FROM PART;
```

```
INSERT INTO MINMAX
SELECT 'SUPPLIER',MIN(S_SUPPKEY),MAX(S_SUPPKEY)
FROM SUPPLIER;
```

```
INSERT INTO MINMAX
SELECT 'PARTSUPP_PART',MIN(PS_PARTKEY),MAX(PS_PARTKEY)
FROM PARTSUPP;
```

```
INSERT INTO MINMAX
SELECT 'PARTSUPP_SUPP',MIN(PS_SUPPKEY),MAX(PS_SUPPKEY)
FROM PARTSUPP ;
```

```
INSERT INTO MINMAX
SELECT 'NATION',MIN(N_NATIONKEY),MAX(N_NATIONKEY)
FROM NATION;
```

```
INSERT INTO MINMAX
SELECT 'REGION',MIN(R_REGIONKEY),MAX(R_REGIONKEY)
FROM REGION;
```

```
SELECT * FROM MINMAX;
spool off
exit;
```

firstten.sql

```
set echo on
set numwidth 25
spool count.out
select * from lineitem where rownum < 11;
select * from orders where rownum < 11;
select * from part where rownum < 11;
select * from partsupp where rownum < 11;
select * from supplier where rownum < 11;
select * from customer where rownum < 11;
select * from nation where rownum < 11;
select * from region where rownum < 11;
spool off
exit;
```

Appendix G - Pricing

No: 062602/ES45

Description	Part Number	Qty	Unit Cost	Ext. HW Cost	Ext. SW Cost
Server Hardware					
Compaq AS ES45 68/1000 M2 4GB Unix	DA-68DAA-DA	4	\$50,809	\$203,236	
3YR 24x7/4HR ES45 M2	FM-4V724-36	4	\$17,896		\$71,584
ES45 Tower Enclosure	BA61M-CT	4	\$325	\$1,300	
ES45 68/1000 SMP CPU Unix	KN610-DB	12	\$10,400	\$124,800	
ES45 4GB Memory Option	MS620-DA	28	\$17,306	\$484,568	
Optional Power Supply, Self Sensing	H7906-A9	4	\$813	\$3,252	
18.2GB 10K RPM Ultra3 SCSI	3R-A0585-AA	13	\$207	\$2,691	
Power Cord	BN18J-1K	8	\$8	\$64	
SCSI Drive Cage, 6 Slot	BA610-6D	4	\$488	\$1,952	
1 CH Wide Ultra-2 (LVD) Adapter	3X-KZPCA-AA	4	\$246	\$984	
PCI to Dual-Prt FE TX NIC	3X-DE602-AA	4	\$246	\$984	
Memory Channel Hub (w/4 line cards)	CCMHB-AA	1	\$6,365	\$6,365	
PCI MC2 Controller	CCMAB-AA	4	\$1,944	\$7,776	
10 Meter Cable for MC2	BN39B-10	4	\$273	\$1,092	
64 Bit FC HBA T64/VMS	DS-KGPSA-CA	16	\$2,209	\$35,344	
D Shelf 180W 1Doc BLW Metric Blue	DS-BA356-JD	2	\$919	\$1,838	
PCI to UltraSCSI Adapter UWSE	KZPBA-CA	2	\$299	\$598	
Ultra 68VHD 3M Cable Assembly	BN37A-03	2	\$91	\$182	
VT510;White; North Amer; No key	VT510-AA	4	\$397	\$1,588	
US/CANADA W95 KYBD WHIT	PCXLA-NA	4	\$16	\$64	
Subtotal				\$878,678	\$71,584
STORAGE:					
USA Model 914 Storage Shelf	DS-SWXEB-AA	4	\$46,406	\$185,624	
Controller w/o ECB Cable Kit	DS-HSG80-BK	24	\$8,793	\$211,032	
16-Port SAN Switch	DS-DSGGB-AB	4	\$18,525	\$74,100	
Short-Wave Optical GBICs	DS-DXGGA-SA	40	\$241	\$9,640	
Model 2200 ECB	DS-SE2CS-CB	24	\$369	\$8,856	
3YR 7X24/4HR EMPTY 4214 NO DRIVES	FM-4E724-36	36	\$126		\$4,536
3YR 7X24/4HR M2 CNTLR SHELF W/CACHE	FM-CK724-36	24	\$407		\$9,768
3YR 7X24/4HR, HSG80 W/CACHE	FM-C9724-36	24	\$2,255		\$54,120
Fibre Channel Cables	BNGBX-30	40	\$246	\$9,840	
12/24gb 4MM Dat 5.25 Tape Drive	TLZ10-LB	4	\$444	\$1,776	
3YR 7X24/4HR,4MM DAT TAPE DRV	FM-4M724-36	4	\$862		\$3,448
9.1GB 7200RPM Disk**	DS-RZ1DA-VW	6	\$423	\$2,538	
18.2GB 10K RPM Ultra3 SCSI	3R-A0585-AA	317	\$206	\$65,302	
Subtotal				\$568,708	\$71,872
Software					
3YR, AS ES40/45 UNIX BRNZ24X7	FM-E4WUS-36	4	\$1,896		\$7,584
3YR Digital Unix O/S & LP	FM-CDDST-36	4	\$6,765		\$27,060
TRU64 UNIX AlphaCDROM	QA-MT4AA-H8	4	\$293	\$1,172	
3YR AS ES45 UNIX SMP	FM-62USM-36	12	\$214		\$2,568
3YR 7X24 HS*80 Platform SW	FM-PLAT2-36	24	\$414		\$9,936
HSG80 ACSsf All Lic/PCRM Pkg	QB-6BUAA-SB	24	\$4,550	\$109,200	
Subtotal				\$110,372	\$47,148
GRAND TOTAL:				\$1,557,758	\$190,604

The Compaq AlphaServer ES45 carries a three (3) year, 7x24HR response warranty. The storage products have a three (3) year on-site, 4-hour, 7-day per week response with a three (3) year return to manufacture warranty. All other products carry a standard warranty of three (3) years on-site: 4-hour x 7-days per week.

Valid: This quote is valid for 60days from date

Terms: TBD

Delivery: 15 Days ARO

Shipping: FOB Origin

Warranty: Manufactures New Equipment

Installation: Included

Sincerely,

Philip K. Nolan