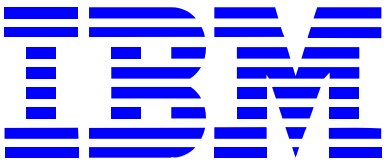


TPC Benchmark™ H Full Disclosure Report

IBM @server p5 595

Using

Oracle Database 10g Enterprise Edition, Release 2 with Partitioning

First Edition

September 19, 2005

The following terms used in this publication are trademarks of their respective companies as follows:

Trademark of the Transaction Processing Performance Council:
TPC Benchmark
TPC-H
QppH
QthH
QphH

Trademark of International Business Machines Corporation:
IBM
the IBM logo
@server
pSeries
POWER5
TotalStorage
AIX
AIX 5L

Trademark of Oracle Corporation:
Oracle 10g
SQL*DBA
SQL*Loader
SQL*Plus
Pro*C
PL/SQL

First Edition September 19, 2005

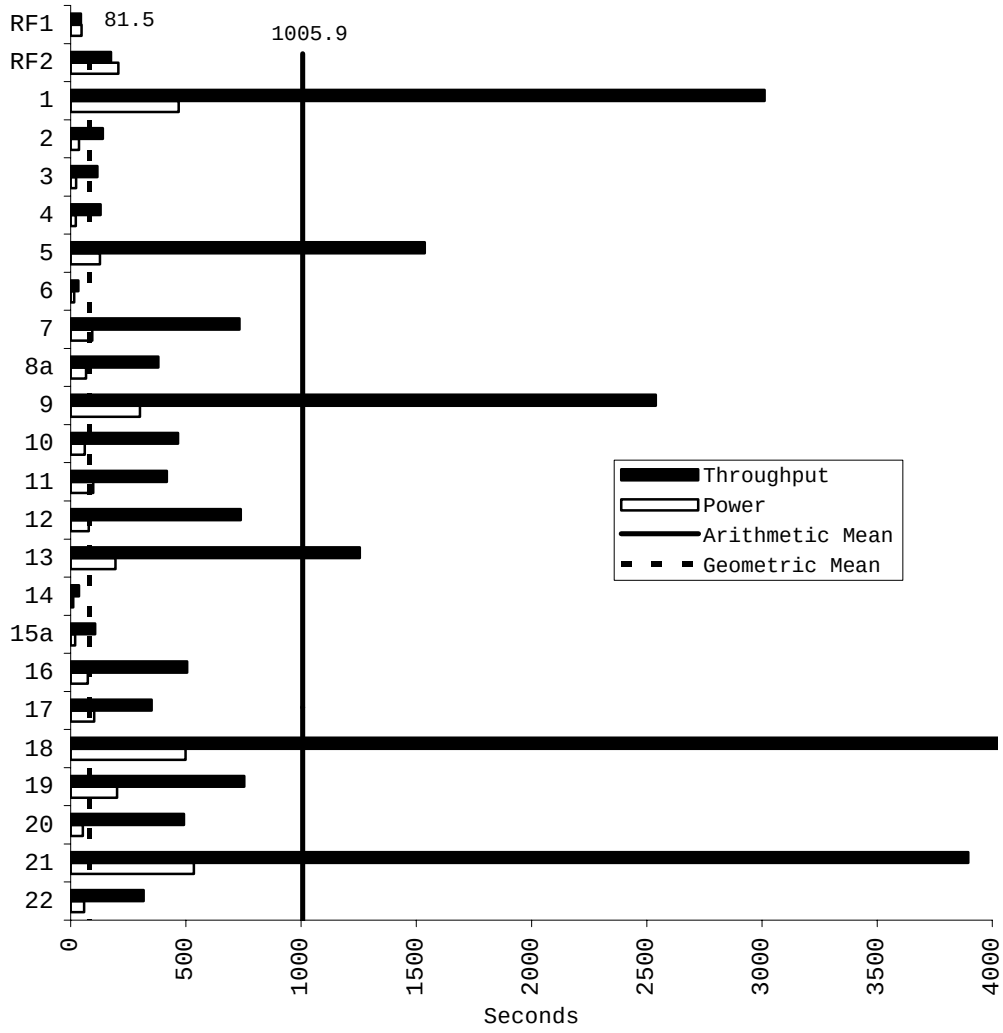
IBM @server p5 595 with Oracle Database 10g Enterprise Edition, Release 2

TPC-H Rev. 2.3.0

Report Date: September 19, 2005

Total System Cost	Composite Query per Hour Rating	Price/Performance
5,358,874 USD	100,512.3 QphH@3000GB	53 USD Price/QphH@3000GB

Database Size	Database Manager	Operating System	Other Soft-	Availability Date
3000GB	Oracle Database 10g Enterprise Edition, Release 2 with Partitioning	AIX 5L V5.3	None	March 1, 2006



Database Load Time: 4:06:30	Load included backup: N	Total Data Storage/Database Size: 12.861
RAID (base tables): Y	RAID (Base Tables and Auxiliary Data Structures): Y	RAID (All): Y

System Configurations

1 IBM @server p5 595 server with

Processors	64 x 1900MHz POWER5 processors on 32 chips, 64 cores, 128 threads, with 32 x 36MB L3 Cache
Memory	256GB
Disk Controllers	96 x 2Gigabit Fibre Channel PCI-X Adapter & 24 x IBM TotalStorage DS4500 dual controllers
Disk Drives	1,152 x 36.4GB 15K 2Gb FC drives ; 4 x 36.4GB 15K Ultra4 SCSI drives
Network	
Total Disk Storage	38,583.5 GB (GB is defined as 1024 * 1024 * 1024 bytes)

IBM @server p5 595 with Oracle Database 10g Enterprise Edition, Release 2

TPC-H Rev. 2.3.0

Report Date: September 19, 2005

Description	Part No.	Source	Unit Price	Qty	Ext Price	Maint Price
Server Hardware						
Server 1:- p5 595	9119-595	1	92,000	1	92,000	63,000
RIO-2 (Remote I/O-2) Cable, 3.5M	3147	1	660	8	5,280	
RIO-2(Remote I/O-2)Cbl, 8.0M	3170	1	800	24	19,200	
36.4 GB 15,000 RPM Ultra320 SCSI Disk Drive	3277	1	2,125	4	8,500	
IBM 10/100/1000 Base-TX Ethernet PCI-X Adapter	5701	1	1,280	1	1,280	
PCI-X Dual Ultra320 SCSI by Adaptec, Inc.	5710	1	790	1	790	
2 Gigabit Fibre Channel PCI-X Adapter	5716	1	2,720	96	261,120	
Expansion Rack, Powered	5792	1	60,000	1	60,000	3,960
I/O Drawer, 20 Slots, 8 Disk Bays	5794	1	30,000	8	240,000	32,640
I/O Drw.Cbl.Grp, Prim.Rck/5U	6122	1	400	1	400	
I/O Drw.Cbl.Grp, Prim.Rck/1U	6123	1	400	1	400	
Bulk Power Regulator	6186	1	4,000	10	40,000	
Slim Line Doors, System and #5792 Racks	6251	1	6,000	2	12,000	
Ethernet Cable, 15M, HMC to System Unit	7802	1	31	3	93	
Power Controller Assembly	7803	1	4,000	4	16,000	
Cooling Group, 2-4 Processor Books	7807	1	4,000	1	4,000	
DC Power Converter, Processor Book	7809	1	6,000	12	72,000	
Processor Clock Card, Programmable	7810	1	575	2	1,150	
System Service Processor	7811	1	3,500	2	7,000	
Multiplexer Card	7812	1	2,200	4	8,800	
16-Way POWER5 Turbo Processor, 0-Way Active	7813	1	127,000	4	508,000	68,640
Activation, #7813 Processor, One Processor	7815	1	32,000	64	2,048,000	506,880
Remote I/O-2 (RIO-2) Loop Adapter, Two Port	7818	1	3,400	16	54,400	
Pwr.Cbl.Grp, CEC Primary Fans	7821	1	650	1	650	
Pwr.Cbl.Grp, 1st CEC Book	7822	1	650	1	650	
Pwr.Cbl.Grp, 2nd CEC Book	7823	1	650	1	650	
Pwr.Cbl.Grp, 3rd CEC Book	7824	1	650	1	650	
Pwr.Cbl.Grp, 4th CEC Book	7825	1	650	1	650	
Pwr.Cbl.Grp, 7807 Cooling Grp.	7826	1	650	1	650	
Power Distribution Assembly	7837	1	2,500	6	15,000	
I/O Drw.Cbl.Grp, 5792 Rck/1U	7853	1	650	1	650	
I/O Drw.Cbl.Grp, 5792 Rck/5U	7854	1	650	1	650	
I/O Drw.Cbl.Grp, 5792 Rck/9U	7855	1	650	1	650	
I/O Drw.Cbl.Grp, Rck/19U	7857	1	400	1	400	
I/O Drw.Cbl.Grp, Rck/23U	7858	1	400	1	400	
I/O Drw.Cbl.Grp, Rck/23U	7859	1	400	1	400	
256GB DDR1 Memory (32 X 8GB Cards)	8195	1	786,432	1	786,432	
Line Cord, 6AWG, 14ft, IEC309 100A Plug	8686	1	2,000	4	8,000	
IBM DVD RAM Drive	7210-025	1	1,400	1	1,400	360
HMC 1:7310-C04 Desktop Hardw.Mgmt.Console	7310-C04	1	1,830	1	1,830	960
10/100 Mbps Ethernet PCI Adapter II	4962	1	412	1	412	
IBM ThinkVision C170 17-inch Color Monitor	3631	1	250	1	250	
Ethernet Cable,6M, HMC to System Unit	7801	1	12	1	12	
Keyboard	8800	1	83	1	83	
Mouse	8841	1	62	1	62	
Power Cable	6470	1	14	1	14	
				Subtotal	**4,280,958**	**676,440**

IBM @server p5 595 with Oracle Database 10g Enterprise Edition, Release 2

TPC-H Rev. 2.3.0

Report Date: September 19, 2005

Storage						
DS4500 Disk System	1742-90U	1	49,900	24	1,197,600	
DS4500 AIX Host Kit	7111	1	4,000	1	4,000	
DS4000 EXP710 Storage Expansion	1740-710	1	6,000	96	576,000	
2Gb FC, 36.4GB/15K Drive	5212	1	1,115	1,152	1,284,480	
Short Wave SFP	2210	1	499	480	239,520	
Fiber Cable 25m	5625	1	189	96	18,144	
Fiber Cable 1m	5601	1	79	192	15,168	
2Gb Mini HUB	3507	1	899	96	86,304	
IBM S2 42U Standard Rack	93074SX	1	1,489	10	14,890	
3-Year 24x7x4 ServicePac for (Rack)	41L2760	1	300	10		3,000
1740-710 24x7x4 WSU	6942-25B	1	760	96		72,960
1742-90U 24x7x4 WSU	6942-25B	1	1,087	24		26,088
				Subtotal	**3,436,106**	**102,048**
Server Software						
AIX 5.3 (media only)	5692-A5L	1	50	1	50	
AIX Software per Processor	5765-G03	1	2,495	64	159,680	
Software Maintenance for AIX, 3 Year	5773-SM3	1	NC	1		
AIX Software Maintanance (3Y per CPU)	0474	1	2,836	64		181,504
AIX Software Maintanance 24x7 Upgrade (3Y Per CPU)	0476	1	732	64		46,848
Virtual I/O Server SW Maintenance: 3 Yr	5773-VI3	1	NC			
Per Processor H5 VIO 3 Yr Maintenance	0781	1	155	64		9,920
Per Processor H5 VIO 3 Yr Maint 24x7 Support	0782	1	64	64		4,096
PLM Per processor 3 Yr SW Maint 24x7 Supporrt	5773-PL3	1	NC			
5773-PL3 SW MAINT 3Y Reg H5	0779	1	35	64		2,240
Per processor 3 Yr SW Maint 24x7 Support H5	0780	1	14	64		896
HMC Initial Software Support 3 Year	5773-RS3	1	NC			
Per Processor Software Support 3 Year	0569	1	675	1		675
Per Processor 24x7 Software Support 3 Year	0570	1	236	1		236
C for AIX user Lic+SW maint 12 MO	D5A1DLL	1	515	1	515	
C for AIX user annual SW maint renewal	E1A1FLL	1	103	2		206
Oracle Database 10g Enterprise Edition Release 2, Named User Plus for 3 years		2	10,000	48 ‡	480,000	
Oracle Partitioning, Named User Plus for 3 years		2	2,500	48 ‡	120,000	
Oracle Mandatory E-Business Discount		2			-121,200	
Oracle Database Server Support Package for 3 years		2	2,000	3		6,000
				Subtotal	**639,045**	**252,621**
				Total	**8,356,109**	**1,031,109**
					IBM Total System Discounts†	**-4,028,344**
†Discounts are based on US list prices for similar					**Three-Year Cost of Ownership**	**5,358,874**
quantities & configurations including pre-payment for maintenance					**QphH@3000**	**100,512.3**
Pricing Sources:					**$/QphH@3000**	**53**

1 IBM: Alycia E. Waller awaller@us.ibm.com (770-835-6612)

2 Oracle: MaryBeth Pierantoni, mary.beth.pierantoni@oracle.com (916-315-5081)

‡ 48 = 0.75 * 64. Explanation: For the purposes of counting the number of processors which require licensing, a multicore chip with "n" cores shall be determined by multiplying "n" cores by a factor of .75.

Audited by: Francois Raab of Infosizing (www.infosizing.com)

The system as configured for the test will be available March 1, 2006.

Numerical Quantities Summary

Measurement Results

Database Scaling (SF/Size)	=	3000
Total Data Storage/Database Size	=	12.861
Start of Database Load	=	9/6/2005 21:34:21
End of Database Load	=	9/7/2005 01:40:51
Database Load Time	=	4:06:30
Query Streams for Throughput Test	=	8
TPC-H Power Metric (QppH@3000GB)	=	132,598.2
TPC-H Throughput Metric (QthH@3000GB)	=	76,190.5
Composite Query-per-Hour Rating (QphH@3000GB)	=	100,512.3
Total System Price over 3 years	=	5,358,874 USD
TPC-H Price/Performance Metric ($/QphH@3000GB)	=	53 USD

Measurement Intervals

Measurement Interval in Throughput Test (Ts) = 24,948 seconds

Duration of Stream Execution

Stream Id	Seed Used	Start Date & Time		End Date & Time		Duration
Stream 00	0907014051	9/7/2005	1:57:09	9/7/2005	2:53:36	0:56:27
Stream 01	0907014052	9/7/2005	2:53:50	9/7/2005	9:13:28	6:19:38
Stream 02	0907014053	9/7/2005	2:53:50	9/7/2005	9:04:16	6:10:26
Stream 03	0907014054	9/7/2005	2:53:50	9/7/2005	9:03:36	6:09:46
Stream 04	0907014055	9/7/2005	2:53:50	9/7/2005	8:39:02	5:45:12
Stream 05	0907014056	9/7/2005	2:53:50	9/7/2005	9:10:44	6:16:54
Stream 06	0907014057	9/7/2005	2:53:50	9/7/2005	9:00:45	6:06:55
Stream 07	0907014058	9/7/2005	2:53:50	9/7/2005	8:48:56	5:55:06
Stream 08	0907014059	9/7/2005	2:53:50	9/7/2005	9:20:30	6:26:40
Refresh		9/7/2005	9:20:30	9/7/2005	9:49:38	0:29:08
Throughput		9/7/2005	2:53:50	9/7/2005	9:49:38	6:55:48

Timing Interval (in seconds)

Stream ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8a	Q9	Q10	Q11	Q12
Stream 0	467.9	36.0	23.7	22.1	126.3	15.0	93.8	66.4	299.9	60.1	97.1	78.5
Stream 01	1762.5	146.3	147.3	24.4	1483.6	44.4	563.5	385.1	2692.9	524.4	305.2	795.0
Stream 02	3385.9	122.6	175.8	158.5	1964.8	23.4	546.4	392.0	2608.9	479.4	595.3	774.4
Stream 03	3133.3	167.4	38.8	142.8	1862.6	25.6	675.7	405.3	2003.0	421.6	827.8	543.3
Stream 04	3316.8	102.0	130.2	167.2	614.2	69.0	698.7	339.7	2263.6	437.9	450.3	899.7
Stream 05	3388.9	138.4	84.1	123.6	1589.9	22.2	760.8	530.2	1555.3	543.0	302.7	778.6
Stream 06	2994.9	175.5	149.3	155.4	1585.5	24.4	744.6	367.2	2718.9	317.1	465.7	670.7
Stream 07	2318.8	102.2	101.4	126.3	1577.3	28.9	1269.6	340.0	3493.0	589.0	295.3	554.4
Stream 08	3789.8	158.4	103.1	134.0	1615.7	24.8	601.9	283.4	2976.2	409.8	94.8	892.5
Minimum	1762.5	102.0	38.8	24.4	614.2	22.2	546.4	283.4	1555.3	317.1	94.8	543.3
Average	3011.4	139.1	116.3	129.0	1536.7	32.8	732.6	380.4	2539.0	465.3	417.1	738.6
Maximum	3789.8	175.5	175.8	167.2	1964.8	69.0	1269.6	530.2	3493.0	589.0	827.8	899.7

Stream ID	Q13	Q14	Q15a	Q16	Q17	Q18	Q19	Q20	Q21	Q22	RF1	RF2
Stream 0	193.9	11.0	19.5	74.7	102.1	498.0	201.4	52.9	534.3	58.1	47.0	206.0
Stream 01	1545.9	29.7	139.7	518.4	426.4	4411.5	454.5	261.8	5684.6	430.9	43.0	88.6
Stream 02	904.1	30.4	95.2	553.2	730.0	3639.9	514.9	1011.5	3197.0	322.1	44.0	185.0
Stream 03	1072.8	21.4	91.5	441.2	269.3	4537.5	1080.6	529.2	3643.7	251.4	44.0	186.0
Stream 04	1563.2	25.7	98.8	456.1	224.0	4229.9	603.6	340.0	3310.3	371.4	44.0	188.0
Stream 05	1006.5	22.0	97.0	481.3	587.2	4371.2	455.0	325.4	5181.9	268.8	44.0	175.0
Stream 06	1409.8	49.1	109.1	568.9	172.6	5087.6	646.9	328.2	2960.5	313.6	43.0	143.0
Stream 07	1239.2	66.1	116.5	538.1	151.2	3033.2	635.7	915.9	3537.3	276.4	42.0	138.0
Stream 08	1289.3	41.7	102.5	487.3	250.9	4130.4	1635.4	223.7	3652.6	301.5	44.0	188.0
Minimum	904.1	21.4	91.5	441.2	151.2	3033.2	454.5	223.7	2960.5	251.4	42.0	88.6
Average	1253.8	35.8	106.3	505.5	351.4	4180.1	753.3	492.0	3896.0	317.0	43.5	161.5
Maximum	1563.2	66.1	139.7	568.9	730.0	5087.6	1635.4	1011.5	5684.6	430.9	44.0	188.0

INFO SIZING

TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL CERTIFIED AUDITOR

Benchmark Sponsor: John J. Makis
IBM eServer Performance
11501 Burnet Road
Austin, TX 78758

September 15, 2005

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

Platform: **IBM @server eServer p5 595**
Database Manager: **Oracle Database 10g Enterprise Edition R2**
Operating System: **AIX 5L V5.3**

The results were:

CPU (Speed)	Memory	Disks	**QphH@3,000GB**
IBM @server eServer p5 595			
64 x POWER5 (1.9 GHz)	32 x 36 MB L3 Cache 256 GB Main	1,152 x 36.4 GB FC 4 x 36.4 GB Ultra4	**100,512.3**

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

- The database records were defined with the proper layout and size
- The database population was generated using DBGEN
- The database was properly scaled to 3,000GB 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

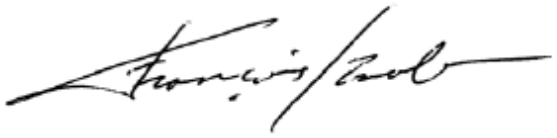
1373 North Franklin Street • Colorado Springs, CO 80903-2527 • Office: 719/473-7555 • Fax: 719/473-7554

- The query input variables were generated by QGEN
- The query text was produced using minor modifications and an approved query variant
- The execution of the queries against the SF1 database produced compliant answers
- A compliant implementation specific layer was used to drive the tests
- The throughput tests involved 8 query streams
- The ratio between the longest and the shortest query was such that no query timing was adjusted
- The execution times for queries and refresh functions were correctly measured and reported
- The repeatability of the measured results was verified
- 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,



François Raab
President

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

Abstract

This report documents the full disclosure information required by the TPC Benchmark™ H Standard Specification Revision 2.3.0 dated August 11, 2005 for measurements on the IBM @server p5 595.

The software used includes AIX 5L V5.3 operating system with Oracle Database 10g Enterprise Edition, Release 2 with Partitioning.

Preface

TPC Benchmark™ H Standard Specification was developed by the Transaction Processing Performance Council (TPC). It was released on February 26, 1999, and most recently revised (Revision 2.3.0) on August 11, 2005. This is the full disclosure report for benchmark testing of the IBM @server p5 595 according to the TPC Benchmark™ H Standard Specification.

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

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

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

- Give answers to real-world business questions;
- Simulate generated ad-hoc queries (e.g., via a point and click GUI interface);
- Are far more complex than most OLTP transactions;
- Include a rich breadth of operators and selectivity constraints;
- Generate intensive activity on the part of the database server component of the system under test;
- Are executed against a database complying to specific population and scaling requirements;
- Are implemented with constraints derived from staying closely synchronized with an on-line production database.

The TPC-H operations are modeled as follows:

- The database is continuously available 24 hours a day, 7 days a week, for ad-hoc queries from multiple end users and data modifications against all tables, except possibly during infrequent (e.g., once a month) maintenance sessions;
- The TPC-H database tracks, possibly with some delay, the state of the OLTP database through on-going refresh functions which batch together a number of modifications impacting some part of the decision support database;
- Due to the worldwide nature of the business data stored in the TPC-H database, the queries and the refresh functions may be executed against the database at any time, especially in relation to each other. In addition, this mix of queries and refresh functions is subject to specific ACIDity requirements, since queries and refresh functions may execute concurrently;

- To achieve the optimal compromise between performance and operational requirements, the database administrator can set, once and for all, the locking levels and the concurrent scheduling rules for queries and refresh functions.

The minimum database required to run the benchmark holds business data from 10,000 suppliers. It contains almost ten million rows representing a raw storage capacity of about 1 gigabyte. Compliant benchmark implementations may also use one of the larger permissible database populations (e.g., 100 gigabytes), as defined in Clause 4.1.3.

The performance metric reported by TPC-H is called the TPC-H Composite Query-per-Hour Performance Metric (QphH@Size), and reflects multiple aspects of the capability of the system to process queries. These aspects include the selected database size against which the queries are executed, the query processing power when queries are submitted by a single stream, and the query throughput when queries are submitted by multiple concurrent users. The TPC-H Price/Performance metric is expressed as $/QphH@Size. To be compliant with the TPC-H standard, all references to TPC-H results for a given configuration must include all required reporting components (see Clause 5.4.6). The TPC believes that comparisons of TPC-H results measured against different database sizes are misleading and discourages such comparisons.

The TPC-H database must be implemented using a commercially available database management system (DBMS) and the queries executed via an interface using dynamic SQL. The specification provides for variants of SQL, as implementers are not required to have implemented a specific SQL standard in full.

TPC-H uses terminology and metrics that are similar to other benchmarks, originated by the TPC and others. Such similarity in terminology does not in any way imply that TPC-H results are comparable to other benchmarks. The only benchmark results comparable to TPC-H are other TPC-H results compliant with the same revision.

Despite the fact that this benchmark offers a rich environment representative of many decision support systems, this benchmark does not reflect the entire range of decision support requirements. In addition, the extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-H approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to any other environment are not recommended.

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

Benchmark sponsors are permitted several possible system designs, provided that they adhere to the model described in Clause 6. A full disclosure report (FDR) of the implementation details, as specified in Clause 8, must be made available along with the reported results.

1.0 General Items

1.1. Benchmark Sponsor

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

This benchmark was sponsored by International Business Machines Corporation.

1.2. Parameter Settings

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

- *Data Base 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.*

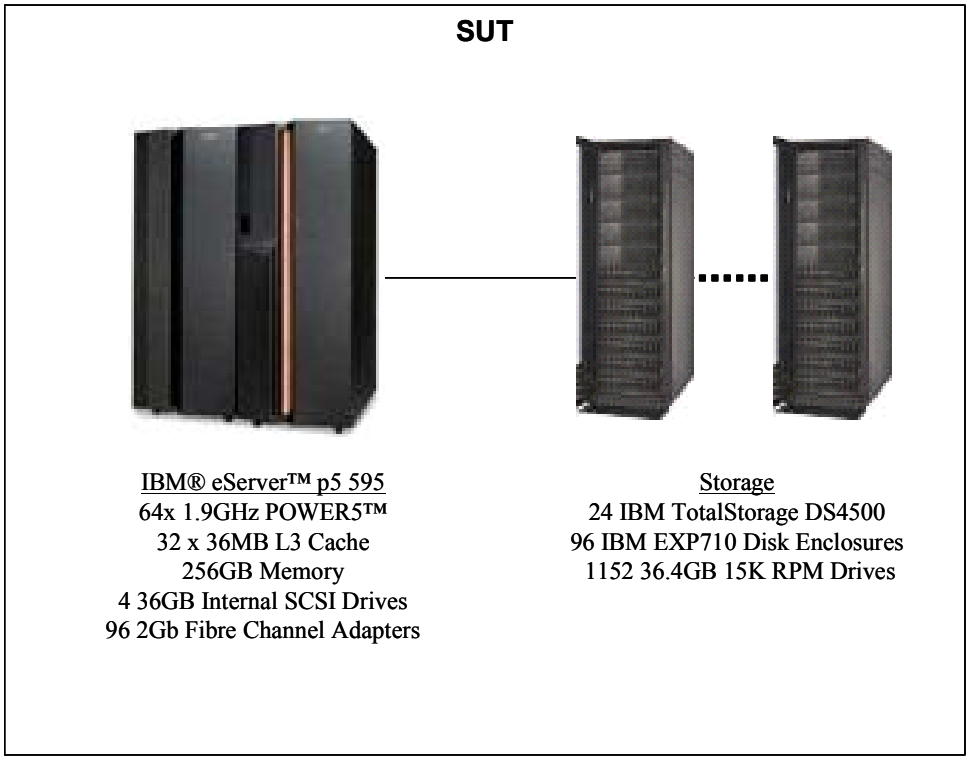
Appendix A "Tunable Parameters" contains a list of Oracle parameters and operating system parameters.

Configuration Diagrams

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

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

IBM @server p5 595 Benchmark Configuration



The system was a single node of IBM @server p5 595 systems with

- 64 x 1900MHz POWER5 processors on 32 chips, 64 cores, 128 threads, with 32 x 36MB L3 Cache
- 256 GB of memory
- 96 IBM 2Gb Fibre Channel PCI-X Adapters
- 4 36.4GB 15K RPM Internal disk drives
- 1,152 36.4GB 15K RPM external disk drives
- 96 EXP710 disk enclosures
- 24 IBM TotalStorage DS4500 dual controllers

For full details of the priced configuration, see the pricing spreadsheet in the Executive Summary.

2.0 Clause 1: Logical Database Design Related Items

Appendix B "Database Build Scripts" contains the programs used to build the test database. The test and qualification databases use the same table definitions, indices and partitioning methods. Thus, the scripts documented in Appendix B were used for both the qualification and test databases except that different file sizes were used to define the log and tablespace devices.

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 "Database Build Scripts" contains the table definitions and the programs to load the database.

2.2. Database Organization

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

Appendix B "Database Build Scripts" contains the DDL for the table and index definitions.

2.3. Horizontal Partitioning

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

Horizontal partitioning was used for all tables except for the nation and region tables, see Appendix B "Database Build Scripts".

2.4. Replication

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

No replication was used.

3.0 Clause 2: Queries and Refresh Functions

3.1. Query Language

The query language used to implement the queries must be identified (e.g.,"RALF/SQL-Plus").

SQL was the query language used.

3.2. Verification for the Random Number Generator

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

The supplied QGEN version 2.1.1b and DBGEN 2.1.1b were used.

3.3. Substitution Parameters

3.3.1. Method of Generation

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

The supplied QGEN version 2.1.1b was used to generate the substitution parameters.

3.4. Query Text

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

Appendix C.1 "Qualification Query Output" contains the output for each of the queries. Variants 8a and 15a were used.

3.5. Query Substitution Parameters and Seeds

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

Appendix F, contains the seed and query substitution parameters.

3.6. Isolation Level

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

The isolation level used to run the queries was repeatable read.

3.7. Refresh Functions

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

The refresh functions are part of the implementation specific layer/driver code included in Appendix D .

4.0 Clause 3: Database System Properties

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

All ACID tests were conducted according to specification. The Atomicity, Consistency, Isolation and Durability tests were performed on the @server p5 595. Appendix E contains the source code for the ACID transaction and query.

4.1. Atomicity Requirements

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

4.1.1. Atomicity of Completed Transaction

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

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

1. The total price from the ORDER table and the extended price from the LINEITEM table were retrieved for a random Orderkey.
2. The ACID transaction was executed for the Orderkey used in Step 1.
3. The ACID transaction committed.
4. The total price and the extended price were retrieved for the same orderkey used in step 1 and step 2. It was verified that the appropriate rows were changed.

4.1.2. Atomicity of Aborted Transactions

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

The following steps were performed to verify the atomicity of the aborted ACID transaction:

1. The total price from the ORDER table and the extended price from the LINEITEM table were retrieved for a random Orderkey.
2. The ACID transaction was executed for the Orderkey used in step 1. The transaction was stopped prior to the commit.
3. The transaction was rolled back.
4. The total price and the extended price were retrieved for the same orderkey used in step 1 and step 2. It was verified that the extended price and the total price were the same as in step 1 and were not changed.

4.2. Consistency Requirements

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

4.2.1. Consistency Tests

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

The queries defined in 4.2.1 , "Consistency Condition" were run after initial database build and prior to executing the ACID transaction. The queries showed that the database is in a consistent state.

After executing 9 streams of 100 ACID transactions each, the queries defined in 4.2.1 , "Consistency Condition" were run again. The queries showed that the database was still in a consistent state.

4.3. Isolation Requirements

4.3.1. Isolation Test 1

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

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

1. An ACID transaction was started with a randomly selected O_KEY, L_KEY and DELTA. The transaction was delayed prior to the Commit.
2. An ACID query was started for the same O_KEY as in the ACID transaction. The ACID query was locked out by the ACID transaction waiting to complete.
3. The ACID transaction was released and the Commit was executed releasing the record.
4. With the LINEITEM record now released, the ACID query can now complete. It returned the data that was committed by the ACID transaction.

4.3.2. Isolation Test 2

This test demonstrates isolation for the read-write conflict of read-write transaction and read-only transaction when the read-write transaction is rolled back.

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

1. An ACID transaction was started for a random O_KEY, L_KEY and DELTA. The transaction was delayed prior to the Rollback.
2. An ACID query was started for the same O_KEY as in the ACID transaction. The ACID query attempts to read the LINEITEM table but was locked out by the ACID transaction.
3. The ACID transaction was released and Rolled back.
4. With the LINEITEM record now released, the ACID query completed.

4.3.3. Isolation Test 3

This test demonstrates isolation for the write-write conflict of two refresh transactions when the first transaction is committed.

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

1. An ACID transaction T1, was started for a randomly selected O_KEY, L_KEY and DELTA. The transaction was delayed prior to the COMMIT.
2. A second ACID transaction T2, was started for the same O_KEY, L_KEY, and for a randomly selected DELTA2. This transaction was forced to wait.
3. The ACID transaction T1 was released and the Commit was executed, releasing the record. With the LINEITEM record now released, the ACID transaction T2 can now complete.
4. Verify that:

   T2.L_EXTENDEDPRICE = T1.L_EXTENDEDPRICE +
   (DELTA*(T1.L_EXTENDEDPRICE/T1.L_QUANTITY))

4.3.4. Isolation Test 4

This test demonstrates isolation for write-write conflict of two ACID transactions when the first transaction is rolled back.

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

1. An ACID transaction T1, was started for a randomly selected O_KEY, L_KEY, and DELTA. The transaction was delayed prior to the rollback.
2. A second ACID transaction T2, was started for the same O_KEY and L_KEY. This transaction was forced to wait.
3. Rolled back the ACID transaction T1. With the LINEITEM record now released, the ACID transaction T2 completed.
4. Verify that T2.L_EXTENDEDPRICE = T1.L_EXTENDEDPRICE

4.3.5. Isolation Test 5

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

1. An ACID transaction T1, was started for a randomly selected O_KEY, L_KEY and DELTA. The ACID transaction was suspended prior to COMMIT.
2. A second ACID transaction T2, was started which selects random values of PS_PARTKEY and PS_SUPPKEY and returned all columns of the PARTSUPP table for which PS_PARTKEY and PS_SUPPKEY were equal to the selected values.
3. T2 completed.
4. T1 was allowed to complete.

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

4.3.6. Isolation Test 6

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

1. A transaction T1, which executes TPC-H query 1 with a randomly selected DELTA, was started.
2. Before T1 completed, an ACID transaction T2, with randomly selected values of O_KEY, L_KEY and DELTA, was started.
3. T2 completed.
4. T1 completed.
5. It was verified that the appropriate rows in the ORDERS, LINEITEM and HISTORY tables were changed.

4.4. Durability Requirements

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.

4.4.1. Permanent Failure of Durable Medium and Loss of System Power

These tests were combined and conducted on the qualification database. The following steps were performed:

1. A test to run 400 ACID transactions on each of 9 execution streams was started such that each stream executes a different set of transactions.
2. One of the disks containing the database transaction log recovery data, database table data, and database index was removed from the enclosure after at least 20 ACID transactions had completed from each of the execution streams.
3. Because the disks were in RAID 1 configuration the applications continued running the ACID transactions.
4. A caching controller which was on the preferred path of database log data, database table data, and database index was removed after at least 50 ACID transactions had completed from each of the execution streams.
5. Because of dual-path redundant mirrored controller the storage system automatically switched the disks to the mirrored controller. The applications continued running the ACID transactions.
6. The system was shutdown by switching off the power for all system components, after at least a total of 100 transactions had completed for each stream.
7. The system was powered back on and rebooted, and the database was restarted and recovered.

5.0 Clause 4: Scaling and Database Population Related Items

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.

The following table contains the TPC Benchmark™ H defined tables and the number of rows for each table as they existed upon build completion:

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

5.2. Distribution of Tables and Logs

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

Oracle was configured on an IBM @server p5 595 server. The system had:

- 96 IBM 2Gb Fibre Channel PCI-X adapters
- 1,152 36.4GB external disk drives.
- 4 36.4GB internal drives

The IBM @server p5 595 was connected to twenty-four IBM TotalStorage DS4500 Storage systems. Each of the DS4500 was configured with 24 RAID-1 arrays containing two disks.

Permanent tables, their auxiliary data structures, database logs, and temporary space resided in tablespaces created on all 576 RAID-1arrays. See Appendix B "Database Build Scripts".

The Operating System resided on an internal disk on the system. The database software resided on an internal disk on the system.

5.3. Mapping of Database Partitions/Replications

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

The database was not replicated.

5.4. Implementation of RAID

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

RAID level 1 was used for database tables, indexes, recovery logs, and temporary spaces.

5.5. DBGEN Modifications

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

The standard distribution of DBGEN version 2.1.1b was used.

5.6. Database Loading

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

The database load time was 4:06:30.

5.7. Data Storage Ratio

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

The calculation of the data storage ratio is shown in the following table:

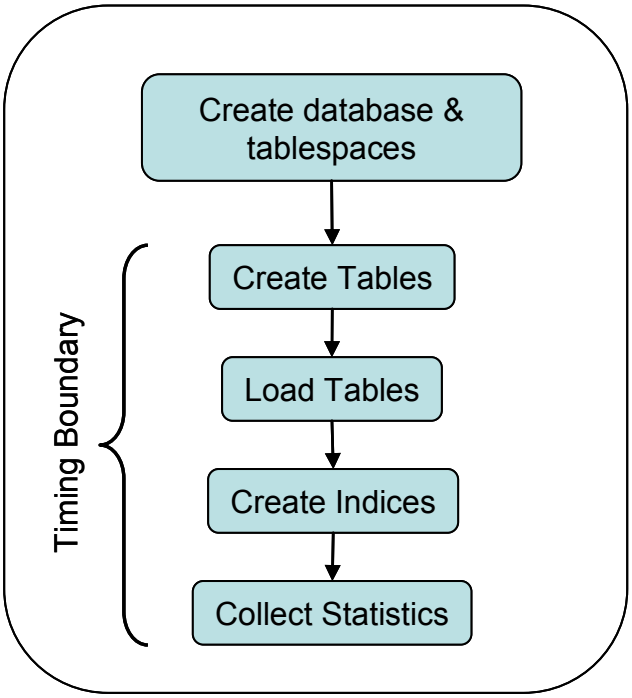
Disk Type	Number of Disks	Space per Disk	Sub-Total Disk Space	Database Size	Data Storage Ratio
ULTRA4 36.4GB	4	33.875 GB	135.5 GB		
2GB FC 36.4GB	1,152	33.375 GB	38,448.00 GB		
Total			38,583.5 GB	3,000GB	12.861

5.8. Details of Database Loading

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

Flat files for each of the tables were created using DBGEN. Appendix B "Database Build Scripts" contains the programs and input files used to load the database.

Database Load Procedure:



6.0 Clause 5: Performance Metrics and Execution-Rules Related Items

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

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

6.2. Steps in the Power Test

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

The following steps were used to implement the power test:

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

6.3. Timing Intervals for Each Query and Refresh Function

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

See Numerical Quantities Summary in the Executive Summary.

6.4. Number of Streams for the Throughput Test

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

See Numerical Quantities Summary in the Executive Summary.

6.5. Start and End Date/Times for Each Query Stream

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

See Numerical Quantities Summary in the Executive Summary.

6.6. Total Elapsed Time for the Measurement Interval

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

See Numerical Quantities Summary in the Executive Summary.

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

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

Stream	Func-tion	Elapsed Time	Start Date	Time	End Date	Time
1	RF1	43	9/7/2005	9:20:30	9/7/2005	9:21:13
	RF2	197	9/7/2005	9:21:13	9/7/2005	9:24:30
2	RF1	44	9/7/2005	9:24:30	9/7/2005	9:25:14
	RF2	185	9/7/2005	9:25:14	9/7/2005	9:28:19
3	RF1	44	9/7/2005	9:28:19	9/7/2005	9:29:03
	RF2	186	9/7/2005	9:29:03	9/7/2005	9:32:09
4	RF1	44	9/7/2005	9:32:09	9/7/2005	9:32:53
	RF2	188	9/7/2005	9:32:53	9/7/2005	9:36:01
5	RF1	44	9/7/2005	9:36:01	9/7/2005	9:36:45
	RF2	175	9/7/2005	9:36:45	9/7/2005	9:39:40
6	RF1	43	9/7/2005	9:39:40	9/7/2005	9:40:23
	RF2	143	9/7/2005	9:40:23	9/7/2005	9:42:46
7	RF1	42	9/7/2005	9:42:46	9/7/2005	9:43:28
	RF2	138	9/7/2005	9:43:28	9/7/2005	9:45:46
8	RF1	44	9/7/2005	9:45:46	9/7/2005	9:46:30
	RF2	188	9/7/2005	9:46:30	9/7/2005	9:49:38

6.8. Timing Intervals for Each Query and Each 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.

See Numerical Quantities Summary in the Executive Summary.

6.9. Performance Metrics

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

See Numerical Quantities Summary in the Executive Summary.

6.10. The Performance Metric and Numerical Quantities from Both Runs

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

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

	QppH@3000GB	QthH@3000GB	QphH@3000GB
Run 1	132,598.2	76,190.5	100,512.3
Run 2	133,920.2	75,629.7	100,639.7

6.11. System Activity between Tests

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

A database checkpoint was initiated with a system log switch.

7.0 Clause 6: SUT and Driver Implementation

7.1. Driver

A detailed textual 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.

Appendix D "Benchmark Scripts" contains the source code used for the driver and all scripts used in connection with it.

The power test and throughput test are performed by the script, runTPCHpt, QGEN is used to generate the query text. The refresh function RF1 is initiated as a separate call with the SQL script runuf1.sh. Then runTPCHpt gets the stream number for each of the streams, and the queries are executed in the order specified in the query file, defined by Clause 5.3.5.4. The refresh function RF2 then is initiated as a separate call with the SQL script, runuf2.sh.

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.

Appendix D “Benchmark Scripts” contains the source code for the qexec utility.

7.3. Profile-Directed Optimization

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

Profile-directed optimization was not used.

8.0 Clause 7: Pricing-Related Items

8.1. Hardware and Software Used

A detailed list of hardware and software used in the priced system must be reported. Each item must have a vendor part number, description, and release/revision level, and indicate General Availability (see Clause 7.2.2.1) either implicitly or explicitly(omitted Availability Dates default to the System Availability 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.

The detailed list of all hardware and software for the priced configuration is listed in the Executive Summary.

8.2. Three Year Cost of System Configuration

The total 3-year price of the entire configuration must be reported, including: hardware, software, hardware maintenance, and software support charges. Separate component pricing is required (see Clause 7.3.1. Pricing Spreadsheet.) Hardware maintenance and software support must be reported separately. The software support level must be disclosed separately from that of hardware, with separate pricing and discounts.

The price sheet for this disclosure is contained in the executive summary pages.

The pricing spreadsheet includes maintenance costs for 3 years. This service provides 7 days per week, 24 hours per day coverage.

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

For assistance with any of these prices or their applicability to any customer's requirements, please contact the sources listed in the pricing spread sheet.

8.3. System Availability Date

The System Availability Date (see Clause 7.2.2.1) must be the single availability date reported on the first page of the executive summary. The full disclosure report must report Availability Dates individually for at least each of the categories for which a pricing subtotal must be provided (see Clause 7.3.1.4). All Availability Dates required to be reported must be disclosed to a precision of 1 day, but the precise format is left to the test sponsor.

The System Availability Date is March 1, 2006.

9.0 Clause 9: Audit Items

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

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

Appendix - A Tunable Parameters

A.1 init.ora

```
AQ_TM_PROCESSES=0
CONTROL_FILES=(+dg1/control01.ctl,+dg1/control02.ctl)
audit_trail = FALSE
compatible=10.2.0.0.0
cpu_count=64
db_block_checksum = false
db_block_size=16384
db_cache_size=38g
db_create_file_dest = +dg1
db_file_multiblock_read_count= 64
db_files=4000
db_name=tpch
db_writer_processes=16
disk_asynch_io=true
dml_locks=80000
global_names=FALSE
instance_name=tpch
job_queue_processes = 0
lock_sga=TRUE
log_buffer =33554432
log_checkpoints_to_alert=true
nls_date_format=YYYY-MM-DD
open_cursors=1024
optimizer_dynamic_sampling = 3
optimizer_features_enable = 10.2.0.1.1
optimizer_index_cost_adj = 160
optimizer_mode = CHOOSE
parallel_adaptive_multi_user = TRUE
parallel_execution_message_size= 32768
parallel_max_servers=1054
parallel_min_servers=384
pga_aggregate_target=75g
processes=4000
query_rewrite_enabled = true
recovery_parallelism = 64
replication_dependency_tracking = FALSE
shared_pool_size=12g
statistics_level = basic
undo_management=AUTO
undo_retention = 200000
```

A.2 init+ASM1.ora

```
instance_type=asm
shared_pool_size=2G
db_cache_size=2g
asm_diskgroups=DG1
+ASM1.instance_number=1
instance_number=1
processes = 500
ASM_DISKSTRING = '/dev/roralv*'
lock_sga=TRUE
```

A.3 AIX Parameters

```
vmo –r –o maxclient% = 15
vmo –r –o maxperm% = 20
vmo –r –o minperm% = 10
vmo –r –o lgpg_regions = 14500
vmo –r –o lgpg_size = 16777216
chdev -l sys0 -a maxuproc=10000


chuser capabilities=CAP_BYPASS_RAC_VMM,CAP_PROPAGATE
,CAP_NUMA_ATTACH oracle
```

A.4 Env

```
########## MACHINE PARAMETERS ###################
########## PATHS ################################
export KIT_DIR=/home/tpch/kit
export SCHEMA_DIR=$KIT_DIR/schema/10.1.0/3000
export PERL=/usr/local/bin/perl
export UTILS=$KIT_DIR/utils
export TEST_DB=/df
export QUAL_DB=$TEST_DB
export DBGEN=$KIT_DIR/dbgen
export ACID_DIR=$KIT_DIR/acid
export QEXEC=$KIT_DIR/utils
export QUERIES=$KIT_DIR/queries
export ANSWERS=$KIT_DIR/answers
export ANS2VAL=/tmp
export ACID_OUT=$QUAL_DB/acid_out
export DSS_CONFIG=$DBGEN
export DSS_QUERY=$KIT_DIR/queries
export DSS_PATH=$DBGEN
export DSS_DIST=dists.dss
export MAINT=$KIT_DIR/maintenance
export FRAME=$KIT_DIR/frame
export SCALE_FACTOR=3000
########### FRAME STUFF
export FRAME_PATH=$KIT_DIR/frame
export ORACORE3INCL=$ORACLE_HOME/rdbms/demo
export ORACORE3PUBL=$ORACLE_HOME/rdbms//public
export RDBMSPUBL=$ORACLE_HOME/rdbms/public
export RDBMSDEMO=$ORACLE_HOME/rdbms/demo
export PLSQLDEMO=$ORACLE_HOME/plsql/demo
export PLSQLPUBL=$ORACLE_HOME/plsql/public
export O=$ORACLE_HOME
export
PATH=./:${BUMPX_DIR}:${UTILS}:${DBGEN}:${MAINT}:${ACID_DIR}
:${FRAME}/bin:${FRAME}/bin:${REGR_TEST}:${PATH}
#
########## ENVIRONMENT VARIABLES #################
export WORKLOAD=TPCH
export HOST=AIX
export OPTLEVEL=X02
export GETOPT=-DSTDLIB_HAS_GETOPT
export PLATFORM=IBM

########## ALIASES ################################

########## RULES - do not change these ############
case "$SCALE_FACTOR" in
  1) export NUM_STREAMS=2;;
  10) export NUM_STREAMS=3;;
  100) export NUM_STREAMS=4;;
  300) export NUM_STREAMS=6;;
  1000) export NUM_STREAMS=7;;
  3000) export NUM_STREAMS=8;;
  10000) export NUM_STREAMS=9;;
esac
export DATABASE_USER=tpch/tpch
```

A.5 Profile

```
USER=${USER:-$LOGNAME}
export USER
export ORACLE_HOME=/home/oracle/OraHome_1
export ORACLE_SID=tpch
export FRAME_PATH=$HOME/tpch/kit/frame
```

Appendix - B Database Build Scripts

B.1 Loadasm

```
#!/bin/ksh

INITFILE=$SCHEMA_DIR/init+ASM1.ora

export ORACLE_SID=+ASM1
sqlplus /NOLOG <<EOF
connect /as sysdba
shutdown abort;
startup pfile= $INITFILE ;
drop diskgroup dg1 including contents;
CREATE DISKGROUP DG1 External REDUNDANCY
DISK
'/dev/roralv1' SIZE 30000M ,
'/dev/roralv2' SIZE 30000M ,
'/dev/roralv3' SIZE 30000M ,
'/dev/roralv4' SIZE 30000M ,
'/dev/roralv5' SIZE 30000M ,
'/dev/roralv6' SIZE 30000M ,
'/dev/roralv7' SIZE 30000M ,
'/dev/roralv8' SIZE 30000M ,
'/dev/roralv9' SIZE 30000M ,
'/dev/roralv10' SIZE 30000M ,
'/dev/roralv11' SIZE 30000M ,
'/dev/roralv12' SIZE 30000M ,
'/dev/roralv13' SIZE 30000M ,
'/dev/roralv14' SIZE 30000M ,
'/dev/roralv15' SIZE 30000M ,
'/dev/roralv16' SIZE 30000M ,
'/dev/roralv17' SIZE 30000M ,
'/dev/roralv18' SIZE 30000M ,
'/dev/roralv19' SIZE 30000M ,
'/dev/roralv20' SIZE 30000M ,
'/dev/roralv21' SIZE 30000M ,
'/dev/roralv22' SIZE 30000M ,
'/dev/roralv23' SIZE 30000M ,
'/dev/roralv24' SIZE 30000M ,
'/dev/roralv25' SIZE 30000M ,
'/dev/roralv26' SIZE 30000M ,
'/dev/roralv27' SIZE 30000M ,
'/dev/roralv28' SIZE 30000M ,
'/dev/roralv29' SIZE 30000M ,
'/dev/roralv30' SIZE 30000M ,
'/dev/roralv31' SIZE 30000M ,
'/dev/roralv32' SIZE 30000M ,
'/dev/roralv33' SIZE 30000M ,
'/dev/roralv34' SIZE 30000M ,
'/dev/roralv35' SIZE 30000M ,
'/dev/roralv36' SIZE 30000M ,
'/dev/roralv37' SIZE 30000M ,
'/dev/roralv38' SIZE 30000M ,
'/dev/roralv39' SIZE 30000M ,
'/dev/roralv40' SIZE 30000M ,
'/dev/roralv41' SIZE 30000M ,
'/dev/roralv42' SIZE 30000M ,
'/dev/roralv43' SIZE 30000M ,
'/dev/roralv44' SIZE 30000M ,
'/dev/roralv45' SIZE 30000M ,
'/dev/roralv46' SIZE 30000M ,
'/dev/roralv47' SIZE 30000M ,
'/dev/roralv48' SIZE 30000M ,
'/dev/roralv49' SIZE 30000M ,
'/dev/roralv50' SIZE 30000M ,
'/dev/roralv51' SIZE 30000M ,
'/dev/roralv52' SIZE 30000M ,
'/dev/roralv53' SIZE 30000M ,
'/dev/roralv54' SIZE 30000M ,
'/dev/roralv55' SIZE 30000M ,
'/dev/roralv56' SIZE 30000M ,
'/dev/roralv57' SIZE 30000M ,
'/dev/roralv58' SIZE 30000M ,
'/dev/roralv59' SIZE 30000M ,
'/dev/roralv60' SIZE 30000M ,
'/dev/roralv61' SIZE 30000M ,
'/dev/roralv62' SIZE 30000M ,
'/dev/roralv63' SIZE 30000M ,
'/dev/roralv64' SIZE 30000M ,
'/dev/roralv65' SIZE 30000M ,
'/dev/roralv66' SIZE 30000M ,
'/dev/roralv67' SIZE 30000M ,
'/dev/roralv68' SIZE 30000M ,
'/dev/roralv69' SIZE 30000M ,
'/dev/roralv70' SIZE 30000M ,
'/dev/roralv71' SIZE 30000M ,
'/dev/roralv72' SIZE 30000M ,
'/dev/roralv73' SIZE 30000M ,
'/dev/roralv74' SIZE 30000M ,
'/dev/roralv75' SIZE 30000M ,
'/dev/roralv76' SIZE 30000M ,
'/dev/roralv77' SIZE 30000M ,
'/dev/roralv78' SIZE 30000M ,
'/dev/roralv79' SIZE 30000M ,
'/dev/roralv80' SIZE 30000M ,
'/dev/roralv81' SIZE 30000M ,
'/dev/roralv82' SIZE 30000M ,
'/dev/roralv83' SIZE 30000M ,
'/dev/roralv84' SIZE 30000M ,
'/dev/roralv85' SIZE 30000M ,
'/dev/roralv86' SIZE 30000M ,
'/dev/roralv87' SIZE 30000M ,
'/dev/roralv88' SIZE 30000M ,
'/dev/roralv89' SIZE 30000M ,
'/dev/roralv90' SIZE 30000M ,
'/dev/roralv91' SIZE 30000M ,
'/dev/roralv92' SIZE 30000M ,
'/dev/roralv93' SIZE 30000M ,
'/dev/roralv94' SIZE 30000M ,
'/dev/roralv95' SIZE 30000M ,
'/dev/roralv96' SIZE 30000M ,
'/dev/roralv97' SIZE 30000M ,
'/dev/roralv98' SIZE 30000M ,
'/dev/roralv99' SIZE 30000M ,
'/dev/roralv100' SIZE 30000M ,
'/dev/roralv101' SIZE 30000M ,
'/dev/roralv102' SIZE 30000M ,
'/dev/roralv103' SIZE 30000M ,
'/dev/roralv104' SIZE 30000M ,
'/dev/roralv105' SIZE 30000M ,
'/dev/roralv106' SIZE 30000M ,
'/dev/roralv107' SIZE 30000M ,
'/dev/roralv108' SIZE 30000M ,
'/dev/roralv109' SIZE 30000M ,
'/dev/roralv110' SIZE 30000M ,
'/dev/roralv111' SIZE 30000M ,
'/dev/roralv112' SIZE 30000M ,
'/dev/roralv113' SIZE 30000M ,
'/dev/roralv114' SIZE 30000M ,
'/dev/roralv115' SIZE 30000M ,
'/dev/roralv116' SIZE 30000M ,
'/dev/roralv117' SIZE 30000M ,
'/dev/roralv118' SIZE 30000M ,
'/dev/roralv119' SIZE 30000M ,
'/dev/roralv120' SIZE 30000M ,
'/dev/roralv121' SIZE 30000M ,
'/dev/roralv122' SIZE 30000M ,
'/dev/roralv123' SIZE 30000M ,
'/dev/roralv124' SIZE 30000M ,
'/dev/roralv125' SIZE 30000M ,
'/dev/roralv126' SIZE 30000M ,
'/dev/roralv127' SIZE 30000M ,
```

```
'/dev/roralv128' SIZE 30000M ,
'/dev/roralv129' SIZE 30000M ,
'/dev/roralv130' SIZE 30000M ,
'/dev/roralv131' SIZE 30000M ,
'/dev/roralv132' SIZE 30000M ,
'/dev/roralv133' SIZE 30000M ,
'/dev/roralv134' SIZE 30000M ,
'/dev/roralv135' SIZE 30000M ,
'/dev/roralv136' SIZE 30000M ,
'/dev/roralv137' SIZE 30000M ,
'/dev/roralv138' SIZE 30000M ,
'/dev/roralv139' SIZE 30000M ,
'/dev/roralv140' SIZE 30000M ,
'/dev/roralv141' SIZE 30000M ,
'/dev/roralv142' SIZE 30000M ,
'/dev/roralv143' SIZE 30000M ,
'/dev/roralv144' SIZE 30000M ,
'/dev/roralv145' SIZE 30000M ,
'/dev/roralv146' SIZE 30000M ,
'/dev/roralv147' SIZE 30000M ,
'/dev/roralv148' SIZE 30000M ,
'/dev/roralv149' SIZE 30000M ,
'/dev/roralv150' SIZE 30000M ,
'/dev/roralv151' SIZE 30000M ,
'/dev/roralv152' SIZE 30000M ,
'/dev/roralv153' SIZE 30000M ,
'/dev/roralv154' SIZE 30000M ,
'/dev/roralv155' SIZE 30000M ,
'/dev/roralv156' SIZE 30000M ,
'/dev/roralv157' SIZE 30000M ,
'/dev/roralv158' SIZE 30000M ,
'/dev/roralv159' SIZE 30000M ,
'/dev/roralv160' SIZE 30000M ,
'/dev/roralv161' SIZE 30000M ,
'/dev/roralv162' SIZE 30000M ,
'/dev/roralv163' SIZE 30000M ,
'/dev/roralv164' SIZE 30000M ,
'/dev/roralv165' SIZE 30000M ,
'/dev/roralv166' SIZE 30000M ,
'/dev/roralv167' SIZE 30000M ,
'/dev/roralv168' SIZE 30000M ,
'/dev/roralv169' SIZE 30000M ,
'/dev/roralv170' SIZE 30000M ,
'/dev/roralv171' SIZE 30000M ,
'/dev/roralv172' SIZE 30000M ,
'/dev/roralv173' SIZE 30000M ,
'/dev/roralv174' SIZE 30000M ,
'/dev/roralv175' SIZE 30000M ,
'/dev/roralv176' SIZE 30000M ,
'/dev/roralv177' SIZE 30000M ,
'/dev/roralv178' SIZE 30000M ,
'/dev/roralv179' SIZE 30000M ,
'/dev/roralv180' SIZE 30000M ,
'/dev/roralv181' SIZE 30000M ,
'/dev/roralv182' SIZE 30000M ,
'/dev/roralv183' SIZE 30000M ,
'/dev/roralv184' SIZE 30000M ,
'/dev/roralv185' SIZE 30000M ,
'/dev/roralv186' SIZE 30000M ,
'/dev/roralv187' SIZE 30000M ,
'/dev/roralv188' SIZE 30000M ,
'/dev/roralv189' SIZE 30000M ,
'/dev/roralv190' SIZE 30000M ,
'/dev/roralv191' SIZE 30000M ,
'/dev/roralv192' SIZE 30000M ,
'/dev/roralv193' SIZE 30000M ,
'/dev/roralv194' SIZE 30000M ,
'/dev/roralv195' SIZE 30000M ,
'/dev/roralv196' SIZE 30000M ,
'/dev/roralv197' SIZE 30000M ,
'/dev/roralv198' SIZE 30000M ,
'/dev/roralv199' SIZE 30000M ,
'/dev/roralv200' SIZE 30000M ,
'/dev/roralv201' SIZE 30000M ,
'/dev/roralv202' SIZE 30000M ,
'/dev/roralv203' SIZE 30000M ,
'/dev/roralv204' SIZE 30000M ,
'/dev/roralv205' SIZE 30000M ,
'/dev/roralv206' SIZE 30000M ,
'/dev/roralv207' SIZE 30000M ,
'/dev/roralv208' SIZE 30000M ,
'/dev/roralv209' SIZE 30000M ,
'/dev/roralv210' SIZE 30000M ,
'/dev/roralv211' SIZE 30000M ,
'/dev/roralv212' SIZE 30000M ,
'/dev/roralv213' SIZE 30000M ,
'/dev/roralv214' SIZE 30000M ,
'/dev/roralv215' SIZE 30000M ,
'/dev/roralv216' SIZE 30000M ,
'/dev/roralv217' SIZE 30000M ,
'/dev/roralv218' SIZE 30000M ,
'/dev/roralv219' SIZE 30000M ,
'/dev/roralv220' SIZE 30000M ,
'/dev/roralv221' SIZE 30000M ,
'/dev/roralv222' SIZE 30000M ,
'/dev/roralv223' SIZE 30000M ,
'/dev/roralv224' SIZE 30000M ,
'/dev/roralv225' SIZE 30000M ,
'/dev/roralv226' SIZE 30000M ,
'/dev/roralv227' SIZE 30000M ,
'/dev/roralv228' SIZE 30000M ,
'/dev/roralv229' SIZE 30000M ,
'/dev/roralv230' SIZE 30000M ,
'/dev/roralv231' SIZE 30000M ,
'/dev/roralv232' SIZE 30000M ,
'/dev/roralv233' SIZE 30000M ,
'/dev/roralv234' SIZE 30000M ,
'/dev/roralv235' SIZE 30000M ,
'/dev/roralv236' SIZE 30000M ,
'/dev/roralv237' SIZE 30000M ,
'/dev/roralv238' SIZE 30000M ,
'/dev/roralv239' SIZE 30000M ,
'/dev/roralv240' SIZE 30000M ,
'/dev/roralv241' SIZE 30000M ,
'/dev/roralv242' SIZE 30000M ,
'/dev/roralv243' SIZE 30000M ,
'/dev/roralv244' SIZE 30000M ,
'/dev/roralv245' SIZE 30000M ,
'/dev/roralv246' SIZE 30000M ,
'/dev/roralv247' SIZE 30000M ,
'/dev/roralv248' SIZE 30000M ,
'/dev/roralv249' SIZE 30000M ,
'/dev/roralv250' SIZE 30000M ,
'/dev/roralv251' SIZE 30000M ,
'/dev/roralv252' SIZE 30000M ,
'/dev/roralv253' SIZE 30000M ,
'/dev/roralv254' SIZE 30000M ,
'/dev/roralv255' SIZE 30000M ,
'/dev/roralv256' SIZE 30000M ,
'/dev/roralv257' SIZE 30000M ,
'/dev/roralv258' SIZE 30000M ,
'/dev/roralv259' SIZE 30000M ,
'/dev/roralv260' SIZE 30000M ,
'/dev/roralv261' SIZE 30000M ,
'/dev/roralv262' SIZE 30000M ,
'/dev/roralv263' SIZE 30000M ,
'/dev/roralv264' SIZE 30000M ,
'/dev/roralv265' SIZE 30000M ,
'/dev/roralv266' SIZE 30000M ,
'/dev/roralv267' SIZE 30000M ,
'/dev/roralv268' SIZE 30000M ,
'/dev/roralv269' SIZE 30000M ,
'/dev/roralv270' SIZE 30000M ,
'/dev/roralv271' SIZE 30000M ,
'/dev/roralv272' SIZE 30000M ,
'/dev/roralv273' SIZE 30000M ,
'/dev/roralv274' SIZE 30000M ,
'/dev/roralv275' SIZE 30000M ,
```

'/dev/roralv276' SIZE 30000M ,
'/dev/roralv277' SIZE 30000M ,
'/dev/roralv278' SIZE 30000M ,
'/dev/roralv279' SIZE 30000M ,
'/dev/roralv280' SIZE 30000M ,
'/dev/roralv281' SIZE 30000M ,
'/dev/roralv282' SIZE 30000M ,
'/dev/roralv283' SIZE 30000M ,
'/dev/roralv284' SIZE 30000M ,
'/dev/roralv285' SIZE 30000M ,
'/dev/roralv286' SIZE 30000M ,
'/dev/roralv287' SIZE 30000M ,
'/dev/roralv288' SIZE 30000M ,
'/dev/roralv289' SIZE 30000M ,
'/dev/roralv290' SIZE 30000M ,
'/dev/roralv291' SIZE 30000M ,
'/dev/roralv292' SIZE 30000M ,
'/dev/roralv293' SIZE 30000M ,
'/dev/roralv294' SIZE 30000M ,
'/dev/roralv295' SIZE 30000M ,
'/dev/roralv296' SIZE 30000M ,
'/dev/roralv297' SIZE 30000M ,
'/dev/roralv298' SIZE 30000M ,
'/dev/roralv299' SIZE 30000M ,
'/dev/roralv300' SIZE 30000M ,
'/dev/roralv301' SIZE 30000M ,
'/dev/roralv302' SIZE 30000M ,
'/dev/roralv303' SIZE 30000M ,
'/dev/roralv304' SIZE 30000M ,
'/dev/roralv305' SIZE 30000M ,
'/dev/roralv306' SIZE 30000M ,
'/dev/roralv307' SIZE 30000M ,
'/dev/roralv308' SIZE 30000M ,
'/dev/roralv309' SIZE 30000M ,
'/dev/roralv310' SIZE 30000M ,
'/dev/roralv311' SIZE 30000M ,
'/dev/roralv312' SIZE 30000M ,
'/dev/roralv313' SIZE 30000M ,
'/dev/roralv314' SIZE 30000M ,
'/dev/roralv315' SIZE 30000M ,
'/dev/roralv316' SIZE 30000M ,
'/dev/roralv317' SIZE 30000M ,
'/dev/roralv318' SIZE 30000M ,
'/dev/roralv319' SIZE 30000M ,
'/dev/roralv320' SIZE 30000M ,
'/dev/roralv321' SIZE 30000M ,
'/dev/roralv322' SIZE 30000M ,
'/dev/roralv323' SIZE 30000M ,
'/dev/roralv324' SIZE 30000M ,
'/dev/roralv325' SIZE 30000M ,
'/dev/roralv326' SIZE 30000M ,
'/dev/roralv327' SIZE 30000M ,
'/dev/roralv328' SIZE 30000M ,
'/dev/roralv329' SIZE 30000M ,
'/dev/roralv330' SIZE 30000M ,
'/dev/roralv331' SIZE 30000M ,
'/dev/roralv332' SIZE 30000M ,
'/dev/roralv333' SIZE 30000M ,
'/dev/roralv334' SIZE 30000M ,
'/dev/roralv335' SIZE 30000M ,
'/dev/roralv336' SIZE 30000M ,
'/dev/roralv337' SIZE 30000M ,
'/dev/roralv338' SIZE 30000M ,
'/dev/roralv339' SIZE 30000M ,
'/dev/roralv340' SIZE 30000M ,
'/dev/roralv341' SIZE 30000M ,
'/dev/roralv342' SIZE 30000M ,
'/dev/roralv343' SIZE 30000M ,
'/dev/roralv344' SIZE 30000M ,
'/dev/roralv345' SIZE 30000M ,
'/dev/roralv346' SIZE 30000M ,
'/dev/roralv347' SIZE 30000M ,
'/dev/roralv348' SIZE 30000M ,
'/dev/roralv349' SIZE 30000M ,
'/dev/roralv350' SIZE 30000M ,
'/dev/roralv351' SIZE 30000M ,
'/dev/roralv352' SIZE 30000M ,
'/dev/roralv353' SIZE 30000M ,
'/dev/roralv354' SIZE 30000M ,
'/dev/roralv355' SIZE 30000M ,
'/dev/roralv356' SIZE 30000M ,
'/dev/roralv357' SIZE 30000M ,
'/dev/roralv358' SIZE 30000M ,
'/dev/roralv359' SIZE 30000M ,
'/dev/roralv360' SIZE 30000M ,
'/dev/roralv361' SIZE 30000M ,
'/dev/roralv362' SIZE 30000M ,
'/dev/roralv363' SIZE 30000M ,
'/dev/roralv364' SIZE 30000M ,
'/dev/roralv365' SIZE 30000M ,
'/dev/roralv366' SIZE 30000M ,
'/dev/roralv367' SIZE 30000M ,
'/dev/roralv368' SIZE 30000M ,
'/dev/roralv369' SIZE 30000M ,
'/dev/roralv370' SIZE 30000M ,
'/dev/roralv371' SIZE 30000M ,
'/dev/roralv372' SIZE 30000M ,
'/dev/roralv373' SIZE 30000M ,
'/dev/roralv374' SIZE 30000M ,
'/dev/roralv375' SIZE 30000M ,
'/dev/roralv376' SIZE 30000M ,
'/dev/roralv377' SIZE 30000M ,
'/dev/roralv378' SIZE 30000M ,
'/dev/roralv379' SIZE 30000M ,
'/dev/roralv380' SIZE 30000M ,
'/dev/roralv381' SIZE 30000M ,
'/dev/roralv382' SIZE 30000M ,
'/dev/roralv383' SIZE 30000M ,
'/dev/roralv384' SIZE 30000M ,
'/dev/roralv385' SIZE 30000M ,
'/dev/roralv386' SIZE 30000M ,
'/dev/roralv387' SIZE 30000M ,
'/dev/roralv388' SIZE 30000M ,
'/dev/roralv389' SIZE 30000M ,
'/dev/roralv390' SIZE 30000M ,
'/dev/roralv391' SIZE 30000M ,
'/dev/roralv392' SIZE 30000M ,
'/dev/roralv393' SIZE 30000M ,
'/dev/roralv394' SIZE 30000M ,
'/dev/roralv395' SIZE 30000M ,
'/dev/roralv396' SIZE 30000M ,
'/dev/roralv397' SIZE 30000M ,
'/dev/roralv398' SIZE 30000M ,
'/dev/roralv399' SIZE 30000M ,
'/dev/roralv400' SIZE 30000M ,
'/dev/roralv401' SIZE 30000M ,
'/dev/roralv402' SIZE 30000M ,
'/dev/roralv403' SIZE 30000M ,
'/dev/roralv404' SIZE 30000M ,
'/dev/roralv405' SIZE 30000M ,
'/dev/roralv406' SIZE 30000M ,
'/dev/roralv407' SIZE 30000M ,
'/dev/roralv408' SIZE 30000M ,
'/dev/roralv409' SIZE 30000M ,
'/dev/roralv410' SIZE 30000M ,
'/dev/roralv411' SIZE 30000M ,
'/dev/roralv412' SIZE 30000M ,
'/dev/roralv413' SIZE 30000M ,
'/dev/roralv414' SIZE 30000M ,
'/dev/roralv415' SIZE 30000M ,
'/dev/roralv416' SIZE 30000M ,
'/dev/roralv417' SIZE 30000M ,
'/dev/roralv418' SIZE 30000M ,
'/dev/roralv419' SIZE 30000M ,
'/dev/roralv420' SIZE 30000M ,
'/dev/roralv421' SIZE 30000M ,
'/dev/roralv422' SIZE 30000M ,
'/dev/roralv423' SIZE 30000M ,

'/dev/roralv424' SIZE 30000M ,
'/dev/roralv425' SIZE 30000M ,
'/dev/roralv426' SIZE 30000M ,
'/dev/roralv427' SIZE 30000M ,
'/dev/roralv428' SIZE 30000M ,
'/dev/roralv429' SIZE 30000M ,
'/dev/roralv430' SIZE 30000M ,
'/dev/roralv431' SIZE 30000M ,
'/dev/roralv432' SIZE 30000M ,
'/dev/roralv433' SIZE 30000M ,
'/dev/roralv434' SIZE 30000M ,
'/dev/roralv435' SIZE 30000M ,
'/dev/roralv436' SIZE 30000M ,
'/dev/roralv437' SIZE 30000M ,
'/dev/roralv438' SIZE 30000M ,
'/dev/roralv439' SIZE 30000M ,
'/dev/roralv440' SIZE 30000M ,
'/dev/roralv441' SIZE 30000M ,
'/dev/roralv442' SIZE 30000M ,
'/dev/roralv443' SIZE 30000M ,
'/dev/roralv444' SIZE 30000M ,
'/dev/roralv445' SIZE 30000M ,
'/dev/roralv446' SIZE 30000M ,
'/dev/roralv447' SIZE 30000M ,
'/dev/roralv448' SIZE 30000M ,
'/dev/roralv449' SIZE 30000M ,
'/dev/roralv450' SIZE 30000M ,
'/dev/roralv451' SIZE 30000M ,
'/dev/roralv452' SIZE 30000M ,
'/dev/roralv453' SIZE 30000M ,
'/dev/roralv454' SIZE 30000M ,
'/dev/roralv455' SIZE 30000M ,
'/dev/roralv456' SIZE 30000M ,
'/dev/roralv457' SIZE 30000M ,
'/dev/roralv458' SIZE 30000M ,
'/dev/roralv459' SIZE 30000M ,
'/dev/roralv460' SIZE 30000M ,
'/dev/roralv461' SIZE 30000M ,
'/dev/roralv462' SIZE 30000M ,
'/dev/roralv463' SIZE 30000M ,
'/dev/roralv464' SIZE 30000M ,
'/dev/roralv465' SIZE 30000M ,
'/dev/roralv466' SIZE 30000M ,
'/dev/roralv467' SIZE 30000M ,
'/dev/roralv468' SIZE 30000M ,
'/dev/roralv469' SIZE 30000M ,
'/dev/roralv470' SIZE 30000M ,
'/dev/roralv471' SIZE 30000M ,
'/dev/roralv472' SIZE 30000M ,
'/dev/roralv473' SIZE 30000M ,
'/dev/roralv474' SIZE 30000M ,
'/dev/roralv475' SIZE 30000M ,
'/dev/roralv476' SIZE 30000M ,
'/dev/roralv477' SIZE 30000M ,
'/dev/roralv478' SIZE 30000M ,
'/dev/roralv479' SIZE 30000M ,
'/dev/roralv480' SIZE 30000M ,
'/dev/roralv481' SIZE 30000M ,
'/dev/roralv482' SIZE 30000M ,
'/dev/roralv483' SIZE 30000M ,
'/dev/roralv484' SIZE 30000M ,
'/dev/roralv485' SIZE 30000M ,
'/dev/roralv486' SIZE 30000M ,
'/dev/roralv487' SIZE 30000M ,
'/dev/roralv488' SIZE 30000M ,
'/dev/roralv489' SIZE 30000M ,
'/dev/roralv490' SIZE 30000M ,
'/dev/roralv491' SIZE 30000M ,
'/dev/roralv492' SIZE 30000M ,
'/dev/roralv493' SIZE 30000M ,
'/dev/roralv494' SIZE 30000M ,
'/dev/roralv495' SIZE 30000M ,
'/dev/roralv496' SIZE 30000M ,
'/dev/roralv497' SIZE 30000M ,
'/dev/roralv498' SIZE 30000M ,
'/dev/roralv499' SIZE 30000M ,
'/dev/roralv500' SIZE 30000M ,
'/dev/roralv501' SIZE 30000M ,
'/dev/roralv502' SIZE 30000M ,
'/dev/roralv503' SIZE 30000M ,
'/dev/roralv504' SIZE 30000M ,
'/dev/roralv505' SIZE 30000M ,
'/dev/roralv506' SIZE 30000M ,
'/dev/roralv507' SIZE 30000M ,
'/dev/roralv508' SIZE 30000M ,
'/dev/roralv509' SIZE 30000M ,
'/dev/roralv510' SIZE 30000M ,
'/dev/roralv511' SIZE 30000M ,
'/dev/roralv512' SIZE 30000M ,
'/dev/roralv513' SIZE 30000M ,
'/dev/roralv514' SIZE 30000M ,
'/dev/roralv515' SIZE 30000M ,
'/dev/roralv516' SIZE 30000M ,
'/dev/roralv517' SIZE 30000M ,
'/dev/roralv518' SIZE 30000M ,
'/dev/roralv519' SIZE 30000M ,
'/dev/roralv520' SIZE 30000M ,
'/dev/roralv521' SIZE 30000M ,
'/dev/roralv522' SIZE 30000M ,
'/dev/roralv523' SIZE 30000M ,
'/dev/roralv524' SIZE 30000M ,
'/dev/roralv525' SIZE 30000M ,
'/dev/roralv526' SIZE 30000M ,
'/dev/roralv527' SIZE 30000M ,
'/dev/roralv528' SIZE 30000M ,
'/dev/roralv529' SIZE 30000M ,
'/dev/roralv530' SIZE 30000M ,
'/dev/roralv531' SIZE 30000M ,
'/dev/roralv532' SIZE 30000M ,
'/dev/roralv533' SIZE 30000M ,
'/dev/roralv534' SIZE 30000M ,
'/dev/roralv535' SIZE 30000M ,
'/dev/roralv536' SIZE 30000M ,
'/dev/roralv537' SIZE 30000M ,
'/dev/roralv538' SIZE 30000M ,
'/dev/roralv539' SIZE 30000M ,
'/dev/roralv540' SIZE 30000M ,
'/dev/roralv541' SIZE 30000M ,
'/dev/roralv542' SIZE 30000M ,
'/dev/roralv543' SIZE 30000M ,
'/dev/roralv544' SIZE 30000M ,
'/dev/roralv545' SIZE 30000M ,
'/dev/roralv546' SIZE 30000M ,
'/dev/roralv547' SIZE 30000M ,
'/dev/roralv548' SIZE 30000M ,
'/dev/roralv549' SIZE 30000M ,
'/dev/roralv550' SIZE 30000M ,
'/dev/roralv551' SIZE 30000M ,
'/dev/roralv552' SIZE 30000M ,
'/dev/roralv553' SIZE 30000M ,
'/dev/roralv554' SIZE 30000M ,
'/dev/roralv555' SIZE 30000M ,
'/dev/roralv556' SIZE 30000M ,
'/dev/roralv557' SIZE 30000M ,
'/dev/roralv558' SIZE 30000M ,
'/dev/roralv559' SIZE 30000M ,
'/dev/roralv560' SIZE 30000M ,
'/dev/roralv561' SIZE 30000M ,
'/dev/roralv562' SIZE 30000M ,
'/dev/roralv563' SIZE 30000M ,
'/dev/roralv564' SIZE 30000M ,
'/dev/roralv565' SIZE 30000M ,
'/dev/roralv566' SIZE 30000M ,
'/dev/roralv567' SIZE 30000M ,
'/dev/roralv568' SIZE 30000M ,
'/dev/roralv569' SIZE 30000M ,
'/dev/roralv570' SIZE 30000M ,
'/dev/roralv571' SIZE 30000M ,

```
'/dev/roralv572' SIZE 30000M ,
'/dev/roralv573' SIZE 30000M ,
'/dev/roralv574' SIZE 30000M ,
'/dev/roralv575' SIZE 30000M ,
'/dev/roralv576' SIZE 30000M ;
alter diskgroup dg1 rebalance power 0;
EOF


sqlplus /NOLOG <<EOF
connect /as sysdba
shutdown normal
EOF


export ORACLE_SID=+ASM1
sqlplus /NOLOG <<EOF
connect /as sysdba
startup pfile= $INITFILE mount
EOF
```

B.2 ci.sql

```
connect /as sysdba
set timing on
set echo on
```

B.3 dbcre_3k.sh

```
#!/bin/ksh

COMMAND="sqlplus  /NOLOG"
#COMMAND="cat"

echo "database creation started at "`date`
dateext=`date | sed s/\ /_/g | sed s/\:/\_/g`

${COMMAND} <<!
connect /as sysdba

shutdown abort;
startup pfile= /home/oracle/tpch/kit/schema/10.1.0/3000/init.ora no-
mount;
create database
  controlfile reuse
  logfile '+DG1' size 31g reuse,
        '+DG1' size 31g reuse
  datafile '+DG1' size 2g reuse
  sysaux datafile '+DG1' size 2g reuse
  undo tablespace ts_undo1
  datafile '+DG1' size 32000M reuse
  maxdatafiles 3000
  maxinstances 2;

set termout off
set echo off
spool /home/oracle/tpch/kit/temp/data_dict_${dateext}.out
@?/rdbms/admin/catalog.sql;
@?/rdbms/admin/catparr.sql;
@?/rdbms/admin/catproc.sql;
connect system/manager
@?/rdbms/admin/utlxplan.sql;
@?/sqlplus/admin/pupbld.sql;
!
```

B.4 sctso_3k.sh

```
#!/bin/ksh
COMMAND="sqlplus  /NOLOG"
#COMMAND="cat"

echo start tablespace creation and alteration at `date`

(( i = 1 ))
while (( i <=18 ))
do
${COMMAND} <<! &
@ci

alter tablespace ts_undo1
  add datafile '+DG1' size 32000M reuse;
!
(( i = $i + 1 ))
done
wait


${COMMAND} <<! &
@ci
drop tablespace ts_default including contents;
create tablespace ts_default
datafile '+DG1' size 12g reuse
extent management local
autoallocate;
!

${COMMAND} <<!
@ci
drop tablespace ts_temp including contents;
create temporary tablespace ts_temp
tempfile '+DG1' size 60G reuse
extent management local
uniform size 5M;
!

(( i = 1 ))
while (( i <=52 ))
do
${COMMAND} <<! &
@ci

alter tablespace ts_temp
  add tempfile '+DG1' size 60G reuse;
!
(( i = $i + 1 ))
done
wait

(( i = 1 ))
while (( i <= 84 ))
do

${COMMAND} <<! &
@ci
drop tablespace l_${i} including contents;
create tablespace l_${i}
datafile '+DG1' size 64512M reuse
extent management local
uniform size 220M
nologging;
!

(( i = $i + 1 ))
done

(( i = 1 ))
while (( i <= 84 ))
do

${COMMAND} <<! &
@ci
drop tablespace o_${i} including contents;
create tablespace o_${i}
datafile '+DG1' size 16G reuse
extent management local
uniform size 64M
```

```
nologging
;
!

(( i = $i + 1 ))
done
wait

${COMMAND} <<! &
@ci
drop tablespace ts_ps including contents;
create tablespace ts_ps
datafile '+DG1' size 32000M reuse
extent management local
autoallocate nologging
;
!

${COMMAND} <<! &
@ci
drop tablespace ts_p including contents;
create tablespace ts_p
datafile '+DG1' size 32000M  reuse
extent management local
autoallocate nologging
;
!

${COMMAND} <<! &
@ci
drop tablespace ts_c including contents;
create tablespace ts_c
datafile '+DG1' size 32000M  reuse
extent management local
autoallocate nologging
;
!

${COMMAND} <<! &
@ci
drop tablespace ts_s including contents;
create tablespace ts_s
datafile '+DG1' size 8G  reuse
extent management local
autoallocate nologging
;
!

${COMMAND} <<! &
@ci
drop tablespace ts_lokey including contents;
create tablespace ts_lokey
datafile '+DG1' size 32000M  reuse
extent management local
autoallocate nologging
;
!

${COMMAND} <<! &
@ci
drop tablespace ts_ookey including contents;
create tablespace ts_ookey
datafile '+DG1' size 32000M  reuse
extent management local
autoallocate nologging
;
!

${COMMAND} <<! &
@ci
drop tablespace ts_supkey including contents;
create tablespace ts_supkey
datafile '+DG1' size 32000M  reuse
extent management local
autoallocate nologging
;
!

${COMMAND} <<! &
@ci
drop tablespace ts_custkey including contents;
create tablespace ts_custkey
datafile '+DG1' size 32000M  reuse
extent management local
autoallocate nologging
;
!
wait
(( i = 1 ))
while (( i <= 13 ))
do

${COMMAND} <<! &
@ci
alter tablespace ts_ps
  add datafile '+DG1' size 32000M reuse;
!

(( i = $i + 1 ))
done

(( i = 1 ))
while (( i <= 3 ))
do

${COMMAND} <<! &
@ci
alter tablespace ts_c
  add datafile '+DG1' size 32000M reuse;
!
(( i = $i + 1 ))

done

(( i = 1 ))
while (( i <= 3 ))
do

${COMMAND} <<! &
@ci
alter tablespace ts_p
  add datafile '+DG1' size 32000M reuse;
!
(( i = $i + 1 ))

done

(( i = 1 ))
while (( i <= 19 ))
do

${COMMAND} <<! &
@ci
alter tablespace ts_lokey
  add datafile '+DG1' size 32000M reuse;
!

(( i = $i + 1 ))
done

(( i = 1 ))
while (( i <= 9 ))
do

${COMMAND} <<! &
@ci
alter tablespace ts_ookey
  add datafile '+DG1' size 32000M reuse;
```

```
!

(( i = $i + 1 ))
done

${COMMAND} <<! &
@ci
alter tablespace ts_supkey
  add datafile '+DG1' size 32000M reuse;
!

${COMMAND} <<! &
@ci
alter tablespace ts_custkey
  add datafile '+DG1' size 32000M reuse;
!

wait

echo End Create Tablespaces at `date`
```

B.5 dapop_3k.sh

```
#!/bin/ksh

echo Start Table Creation at `date`

sqlplus /NOLOG <<!
connect / as sysdba
set timing on
set echo on
set termout on

drop user tpch cascade;
grant DBA
to tpch identified by tpch;

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

connect tpch/tpch
drop directory d1;
create directory d1 as '/flatfiles/flat1/3000';
drop directory d2;
create directory d2 as '/flatfiles/flat2/3000';
drop directory d3;
create directory d3 as '/flatfiles/flat3/3000';
drop directory d4;
create directory d4 as '/flatfiles/flat4/3000';
drop directory d5;
create directory d5 as '/flatfiles/flat5/3000';
drop directory d6;
create directory d6 as '/flatfiles/flat6/3000';
drop directory d7;
create directory d7 as '/flatfiles/flat7/3000';
drop directory d8;
create directory d8 as '/flatfiles/flat8/3000';
drop directory d9;
create directory d9 as '/flatfiles/flat9/3000';
drop directory d10;
create directory d10 as '/flatfiles/flat10/3000';
drop directory d11;
create directory d11 as '/flatfiles/flat11/3000';
drop directory d12;
create directory d12 as '/flatfiles/flat12/3000';
drop directory d13;
create directory d13 as '/flatfiles/flat13/3000';
drop directory d14;
create directory d14 as '/flatfiles/flat14/3000';
drop directory d15;
create directory d15 as '/flatfiles/flat15/3000';
drop directory d16;
create directory d16 as '/flatfiles/flat16/3000';
drop directory d17;
create directory d17 as '/flatfiles/flat17/3000';
drop directory d18;
create directory d18 as '/flatfiles/flat18/3000';
drop directory d19;
create directory d19 as '/flatfiles/flat19/3000';
drop directory d20;
create directory d20 as '/flatfiles/flat20/3000';
drop directory d21;
create directory d21 as '/flatfiles/flat21/3000';
drop directory d22;
create directory d22 as '/flatfiles/flat22/3000';
drop directory d23;
create directory d23 as '/flatfiles/flat23/3000';
drop directory d24;
create directory d24 as '/flatfiles/flat24/3000';
drop directory d25;
create directory d25 as '/flatfiles/flat25/3000';
drop directory d26;
create directory d26 as '/flatfiles/flat26/3000';
drop directory d27;
create directory d27 as '/flatfiles/flat27/3000';
drop directory d28;
create directory d28 as '/flatfiles/flat28/3000';
drop directory d29;
create directory d29 as '/flatfiles/flat29/3000';
drop directory d30;
create directory d30 as '/flatfiles/flat30/3000';
drop directory d31;
create directory d31 as '/flatfiles/flat31/3000';
drop directory d32;
create directory d32 as '/flatfiles/flat32/3000';
drop directory d33;
create directory d33 as '/flatfiles/flat33/3000';
drop directory d34;
create directory d34 as '/flatfiles/flat34/3000';
drop directory d35;
create directory d35 as '/flatfiles/flat35/3000';
drop directory d36;
create directory d36 as '/flatfiles/flat36/3000';
drop directory d37;
create directory d37 as '/flatfiles/flat37/3000';
drop directory d38;
create directory d38 as '/flatfiles/flat38/3000';
drop directory d39;
create directory d39 as '/flatfiles/flat39/3000';
drop directory d40;
create directory d40 as '/flatfiles/flat40/3000';
drop directory d41;
create directory d41 as '/flatfiles/flat41/3000';
drop directory d42;
create directory d42 as '/flatfiles/flat42/3000';
drop directory d43;
create directory d43 as '/flatfiles/flat43/3000';
drop directory d44;
create directory d44 as '/flatfiles/flat44/3000';
drop directory d45;
create directory d45 as '/flatfiles/flat45/3000';
drop directory d46;
create directory d46 as '/flatfiles/flat46/3000';
drop directory d47;
create directory d47 as '/flatfiles/flat47/3000';
drop directory d48;
create directory d48 as '/flatfiles/flat48/3000';
drop directory d49;
create directory d49 as '/flatfiles/flat49/3000';
drop directory d50;
create directory d50 as '/flatfiles/flat50/3000';
drop directory d51;
create directory d51 as '/flatfiles/flat51/3000';
drop directory d52;
create directory d52 as '/flatfiles/flat52/3000';
drop directory d53;
create directory d53 as '/flatfiles/flat53/3000';
```

```
drop directory d54;
create directory d54 as '/flatfiles/flat54/3000';
drop directory d55;
create directory d55 as '/flatfiles/flat55/3000';
drop directory d56;
create directory d56 as '/flatfiles/flat56/3000';
drop directory d57;
create directory d57 as '/flatfiles/flat57/3000';
drop directory d58;
create directory d58 as '/flatfiles/flat58/3000';
drop directory d59;
create directory d59 as '/flatfiles/flat59/3000';
drop directory d60;
create directory d60 as '/flatfiles/flat60/3000';
drop directory d61;
create directory d61 as '/flatfiles/flat61/3000';
drop directory d62;
create directory d62 as '/flatfiles/flat62/3000';
drop directory d63;
create directory d63 as '/flatfiles/flat63/3000';
drop directory d64;
create directory d64 as '/flatfiles/flat64/3000';

drop table lineitem_et;
create table lineitem_et (
   l_orderkey            number ,
    l_partkey            number ,
   l_suppkey             number ,
   l_linenumber          number ,
   l_quantity            BINARY_DOUBLE ,
   l_extendedprice       BINARY_DOUBLE ,
   l_discount            BINARY_DOUBLE ,
   l_tax                 BINARY_DOUBLE ,
   l_returnflag          char(1) ,
   l_linestatus          char(1) ,
   l_shipdate            date ,
   l_commitdate          date ,
   l_receiptdate         date ,
   l_shipinstruct        char(25) ,
   l_shipmo              char(10) ,
   l_comment             varchar(44)
)
organization external (
type ORACLE_LOADER
default directory d1
access parameters
(
      records delimited by newline
      nobadfile
      nologfile
      nodiscardfile
      fields terminated by '|'
      missing field values are null
)
location (
d1:'lineitem.tbl.1',
d2:'lineitem.tbl.2',
d3:'lineitem.tbl.3',
d4:'lineitem.tbl.4',
d5:'lineitem.tbl.5',
d6:'lineitem.tbl.6',
d7:'lineitem.tbl.7',
d8:'lineitem.tbl.8',
d9:'lineitem.tbl.9',
d10:'lineitem.tbl.10',
d11:'lineitem.tbl.11',
d12:'lineitem.tbl.12',
d13:'lineitem.tbl.13',
d14:'lineitem.tbl.14',
d15:'lineitem.tbl.15',
d16:'lineitem.tbl.16',
d17:'lineitem.tbl.17',
d18:'lineitem.tbl.18',
d19:'lineitem.tbl.19',
d20:'lineitem.tbl.20',
d21:'lineitem.tbl.21',
d22:'lineitem.tbl.22',
d23:'lineitem.tbl.23',
d24:'lineitem.tbl.24',
d25:'lineitem.tbl.25',
d26:'lineitem.tbl.26',
d27:'lineitem.tbl.27',
d28:'lineitem.tbl.28',
d29:'lineitem.tbl.29',
d30:'lineitem.tbl.30',
d31:'lineitem.tbl.31',
d32:'lineitem.tbl.32',
d33:'lineitem.tbl.33',
d34:'lineitem.tbl.34',
d35:'lineitem.tbl.35',
d36:'lineitem.tbl.36',
d37:'lineitem.tbl.37',
d38:'lineitem.tbl.38',
d39:'lineitem.tbl.39',
d40:'lineitem.tbl.40',
d41:'lineitem.tbl.41',
d42:'lineitem.tbl.42',
d43:'lineitem.tbl.43',
d44:'lineitem.tbl.44',
d45:'lineitem.tbl.45',
d46:'lineitem.tbl.46',
d47:'lineitem.tbl.47',
d48:'lineitem.tbl.48',
d49:'lineitem.tbl.49',
d50:'lineitem.tbl.50',
d51:'lineitem.tbl.51',
d52:'lineitem.tbl.52',
d53:'lineitem.tbl.53',
d54:'lineitem.tbl.54',
d55:'lineitem.tbl.55',
d56:'lineitem.tbl.56',
d57:'lineitem.tbl.57',
d58:'lineitem.tbl.58',
d59:'lineitem.tbl.59',
d60:'lineitem.tbl.60',
d61:'lineitem.tbl.61',
d62:'lineitem.tbl.62',
d63:'lineitem.tbl.63',
d64:'lineitem.tbl.64',
d1:'lineitem.tbl.65',
d2:'lineitem.tbl.66',
d3:'lineitem.tbl.67',
d4:'lineitem.tbl.68',
d5:'lineitem.tbl.69',
d6:'lineitem.tbl.70',
d7:'lineitem.tbl.71',
d8:'lineitem.tbl.72',
d9:'lineitem.tbl.73',
d10:'lineitem.tbl.74',
d11:'lineitem.tbl.75',
d12:'lineitem.tbl.76',
d13:'lineitem.tbl.77',
d14:'lineitem.tbl.78',
d15:'lineitem.tbl.79',
d16:'lineitem.tbl.80',
d17:'lineitem.tbl.81',
d18:'lineitem.tbl.82',
d19:'lineitem.tbl.83',
d20:'lineitem.tbl.84',
d21:'lineitem.tbl.85',
d22:'lineitem.tbl.86',
d23:'lineitem.tbl.87',
d24:'lineitem.tbl.88',
d25:'lineitem.tbl.89',
d26:'lineitem.tbl.90',
d27:'lineitem.tbl.91',
d28:'lineitem.tbl.92',
d29:'lineitem.tbl.93',
```

```
d30:'lineitem.tbl.94',
d31:'lineitem.tbl.95',
d32:'lineitem.tbl.96',
d33:'lineitem.tbl.97',
d34:'lineitem.tbl.98',
d35:'lineitem.tbl.99',
d36:'lineitem.tbl.100',
d37:'lineitem.tbl.101',
d38:'lineitem.tbl.102',
d39:'lineitem.tbl.103',
d40:'lineitem.tbl.104',
d41:'lineitem.tbl.105',
d42:'lineitem.tbl.106',
d43:'lineitem.tbl.107',
d44:'lineitem.tbl.108',
d45:'lineitem.tbl.109',
d46:'lineitem.tbl.110',
d47:'lineitem.tbl.111',
d48:'lineitem.tbl.112',
d49:'lineitem.tbl.113',
d50:'lineitem.tbl.114',
d51:'lineitem.tbl.115',
d52:'lineitem.tbl.116',
d53:'lineitem.tbl.117',
d54:'lineitem.tbl.118',
d55:'lineitem.tbl.119',
d56:'lineitem.tbl.120',
d57:'lineitem.tbl.121',
d58:'lineitem.tbl.122',
d59:'lineitem.tbl.123',
d60:'lineitem.tbl.124',
d61:'lineitem.tbl.125',
d62:'lineitem.tbl.126',
d63:'lineitem.tbl.127',
d64:'lineitem.tbl.128'
))reject limit unlimited;
alter table lineitem_et parallel;

drop table orders_et;
create table orders_et (
   o_orderkey          number ,
   o_custkey           number ,
   o_orderstatus       char(1) ,
   o_totalprice        BINARY_DOUBLE ,
   o_orderdate         date ,
   o_orderpriority     char(15) ,
   o_clerk             char(15) ,
   o_shippriority      number ,
   o_comment           varchar(79)
)
organization external (
type ORACLE_LOADER
default directory d1
access parameters
(
      records delimited by newline
      nobadfile
      nologfile
      nodiscardfile
      fields terminated by '|'
      missing field values are null
)
location (
d1:'orders.tbl.1',
d2:'orders.tbl.2',
d3:'orders.tbl.3',
d4:'orders.tbl.4',
d5:'orders.tbl.5',
d6:'orders.tbl.6',
d7:'orders.tbl.7',
d8:'orders.tbl.8',
d9:'orders.tbl.9',
d10:'orders.tbl.10',
d11:'orders.tbl.11',
d12:'orders.tbl.12',
d13:'orders.tbl.13',
d14:'orders.tbl.14',
d15:'orders.tbl.15',
d16:'orders.tbl.16',
d17:'orders.tbl.17',
d18:'orders.tbl.18',
d19:'orders.tbl.19',
d20:'orders.tbl.20',
d21:'orders.tbl.21',
d22:'orders.tbl.22',
d23:'orders.tbl.23',
d24:'orders.tbl.24',
d25:'orders.tbl.25',
d26:'orders.tbl.26',
d27:'orders.tbl.27',
d28:'orders.tbl.28',
d29:'orders.tbl.29',
d30:'orders.tbl.30',
d31:'orders.tbl.31',
d32:'orders.tbl.32',
d33:'orders.tbl.33',
d34:'orders.tbl.34',
d35:'orders.tbl.35',
d36:'orders.tbl.36',
d37:'orders.tbl.37',
d38:'orders.tbl.38',
d39:'orders.tbl.39',
d40:'orders.tbl.40',
d41:'orders.tbl.41',
d42:'orders.tbl.42',
d43:'orders.tbl.43',
d44:'orders.tbl.44',
d45:'orders.tbl.45',
d46:'orders.tbl.46',
d47:'orders.tbl.47',
d48:'orders.tbl.48',
d49:'orders.tbl.49',
d50:'orders.tbl.50',
d51:'orders.tbl.51',
d52:'orders.tbl.52',
d53:'orders.tbl.53',
d54:'orders.tbl.54',
d55:'orders.tbl.55',
d56:'orders.tbl.56',
d57:'orders.tbl.57',
d58:'orders.tbl.58',
d59:'orders.tbl.59',
d60:'orders.tbl.60',
d61:'orders.tbl.61',
d62:'orders.tbl.62',
d63:'orders.tbl.63',
d64:'orders.tbl.64',
d1:'orders.tbl.65',
d2:'orders.tbl.66',
d3:'orders.tbl.67',
d4:'orders.tbl.68',
d5:'orders.tbl.69',
d6:'orders.tbl.70',
d7:'orders.tbl.71',
d8:'orders.tbl.72',
d9:'orders.tbl.73',
d10:'orders.tbl.74',
d11:'orders.tbl.75',
d12:'orders.tbl.76',
d13:'orders.tbl.77',
d14:'orders.tbl.78',
d15:'orders.tbl.79',
d16:'orders.tbl.80',
d17:'orders.tbl.81',
d18:'orders.tbl.82',
d19:'orders.tbl.83',
d20:'orders.tbl.84',
d21:'orders.tbl.85',
```

```
d22:'orders.tbl.86',
d23:'orders.tbl.87',
d24:'orders.tbl.88',
d25:'orders.tbl.89',
d26:'orders.tbl.90',
d27:'orders.tbl.91',
d28:'orders.tbl.92',
d29:'orders.tbl.93',
d30:'orders.tbl.94',
d31:'orders.tbl.95',
d32:'orders.tbl.96',
d33:'orders.tbl.97',
d34:'orders.tbl.98',
d35:'orders.tbl.99',
d36:'orders.tbl.100',
d37:'orders.tbl.101',
d38:'orders.tbl.102',
d39:'orders.tbl.103',
d40:'orders.tbl.104',
d41:'orders.tbl.105',
d42:'orders.tbl.106',
d43:'orders.tbl.107',
d44:'orders.tbl.108',
d45:'orders.tbl.109',
d46:'orders.tbl.110',
d47:'orders.tbl.111',
d48:'orders.tbl.112',
d49:'orders.tbl.113',
d50:'orders.tbl.114',
d51:'orders.tbl.115',
d52:'orders.tbl.116',
d53:'orders.tbl.117',
d54:'orders.tbl.118',
d55:'orders.tbl.119',
d56:'orders.tbl.120',
d57:'orders.tbl.121',
d58:'orders.tbl.122',
d59:'orders.tbl.123',
d60:'orders.tbl.124',
d61:'orders.tbl.125',
d62:'orders.tbl.126',
d63:'orders.tbl.127',
d64:'orders.tbl.128'
))reject limit unlimited;
alter table orders_et parallel;

drop table part_et;
create table part_et (
   p_partkey          number ,
   p_name             varchar(55) ,
   p_mfgr             char(25) ,
   p_brand            char(10) ,
   p_type             varchar(25) ,
   p_size             number ,
   p_container        char(10) ,
   p_retailprice      BINARY_DOUBLE ,
   p_comment          varchar(23)
)
organization external (
type ORACLE_LOADER
default directory d1
access parameters
(
      records delimited by newline
      nobadfile
      nologfile
      nodiscardfile
      fields terminated by '|'
      missing field values are null
)
location (
d1:'part.tbl.1',
d2:'part.tbl.2',
d3:'part.tbl.3',
d4:'part.tbl.4',
d5:'part.tbl.5',
d6:'part.tbl.6',
d7:'part.tbl.7',
d8:'part.tbl.8',
d9:'part.tbl.9',
d10:'part.tbl.10',
d11:'part.tbl.11',
d12:'part.tbl.12',
d13:'part.tbl.13',
d14:'part.tbl.14',
d15:'part.tbl.15',
d16:'part.tbl.16',
d17:'part.tbl.17',
d18:'part.tbl.18',
d19:'part.tbl.19',
d20:'part.tbl.20',
d21:'part.tbl.21',
d22:'part.tbl.22',
d23:'part.tbl.23',
d24:'part.tbl.24',
d25:'part.tbl.25',
d26:'part.tbl.26',
d27:'part.tbl.27',
d28:'part.tbl.28',
d29:'part.tbl.29',
d30:'part.tbl.30',
d31:'part.tbl.31',
d32:'part.tbl.32',
d33:'part.tbl.33',
d34:'part.tbl.34',
d35:'part.tbl.35',
d36:'part.tbl.36',
d37:'part.tbl.37',
d38:'part.tbl.38',
d39:'part.tbl.39',
d40:'part.tbl.40',
d41:'part.tbl.41',
d42:'part.tbl.42',
d43:'part.tbl.43',
d44:'part.tbl.44',
d45:'part.tbl.45',
d46:'part.tbl.46',
d47:'part.tbl.47',
d48:'part.tbl.48',
d49:'part.tbl.49',
d50:'part.tbl.50',
d51:'part.tbl.51',
d52:'part.tbl.52',
d53:'part.tbl.53',
d54:'part.tbl.54',
d55:'part.tbl.55',
d56:'part.tbl.56',
d57:'part.tbl.57',
d58:'part.tbl.58',
d59:'part.tbl.59',
d60:'part.tbl.60',
d61:'part.tbl.61',
d62:'part.tbl.62',
d63:'part.tbl.63',
d64:'part.tbl.64'
))reject limit unlimited;
alter table part_et parallel;

drop table partsupp_et;
create table partsupp_et (
   ps_partkey          number ,
   ps_suppkey          number ,
   ps_availqty         number ,
   ps_supplycost       BINARY_DOUBLE ,
   ps_comment          varchar(199)
)
organization external (
type ORACLE_LOADER
```

```
default directory d1
access parameters
(
      records delimited by newline
      nobadfile
      nologfile
      nodiscardfile
      fields terminated by '|'
      missing field values are null
)
location (
d1:'partsupp.tbl.1',
d2:'partsupp.tbl.2',
d3:'partsupp.tbl.3',
d4:'partsupp.tbl.4',
d5:'partsupp.tbl.5',
d6:'partsupp.tbl.6',
d7:'partsupp.tbl.7',
d8:'partsupp.tbl.8',
d9:'partsupp.tbl.9',
d10:'partsupp.tbl.10',
d11:'partsupp.tbl.11',
d12:'partsupp.tbl.12',
d13:'partsupp.tbl.13',
d14:'partsupp.tbl.14',
d15:'partsupp.tbl.15',
d16:'partsupp.tbl.16',
d17:'partsupp.tbl.17',
d18:'partsupp.tbl.18',
d19:'partsupp.tbl.19',
d20:'partsupp.tbl.20',
d21:'partsupp.tbl.21',
d22:'partsupp.tbl.22',
d23:'partsupp.tbl.23',
d24:'partsupp.tbl.24',
d25:'partsupp.tbl.25',
d26:'partsupp.tbl.26',
d27:'partsupp.tbl.27',
d28:'partsupp.tbl.28',
d29:'partsupp.tbl.29',
d30:'partsupp.tbl.30',
d31:'partsupp.tbl.31',
d32:'partsupp.tbl.32',
d33:'partsupp.tbl.33',
d34:'partsupp.tbl.34',
d35:'partsupp.tbl.35',
d36:'partsupp.tbl.36',
d37:'partsupp.tbl.37',
d38:'partsupp.tbl.38',
d39:'partsupp.tbl.39',
d40:'partsupp.tbl.40',
d41:'partsupp.tbl.41',
d42:'partsupp.tbl.42',
d43:'partsupp.tbl.43',
d44:'partsupp.tbl.44',
d45:'partsupp.tbl.45',
d46:'partsupp.tbl.46',
d47:'partsupp.tbl.47',
d48:'partsupp.tbl.48',
d49:'partsupp.tbl.49',
d50:'partsupp.tbl.50',
d51:'partsupp.tbl.51',
d52:'partsupp.tbl.52',
d53:'partsupp.tbl.53',
d54:'partsupp.tbl.54',
d55:'partsupp.tbl.55',
d56:'partsupp.tbl.56',
d57:'partsupp.tbl.57',
d58:'partsupp.tbl.58',
d59:'partsupp.tbl.59',
d60:'partsupp.tbl.60',
d61:'partsupp.tbl.61',
d62:'partsupp.tbl.62',
d63:'partsupp.tbl.63',
d64:'partsupp.tbl.64'
))reject limit unlimited;
alter table partsupp_et parallel;

drop table supplier_et;
create table supplier_et (
   s_suppkey          number ,
   s_name             char(25) ,
   s_address          varchar(40) ,
   s_nationkey        number ,
   s_phone            char(15) ,
   s_acctbal          BINARY_DOUBLE ,
   s_comment          varchar(101)
)
organization external (
type ORACLE_LOADER
default directory d1
access parameters
(
      records delimited by newline
      nobadfile
      nologfile
      nodiscardfile
      fields terminated by '|'
      missing field values are null
)
location (
d1:'supplier.tbl.1',
d2:'supplier.tbl.2',
d3:'supplier.tbl.3',
d4:'supplier.tbl.4',
d5:'supplier.tbl.5',
d6:'supplier.tbl.6',
d7:'supplier.tbl.7',
d8:'supplier.tbl.8',
d9:'supplier.tbl.9',
d10:'supplier.tbl.10',
d11:'supplier.tbl.11',
d12:'supplier.tbl.12',
d13:'supplier.tbl.13',
d14:'supplier.tbl.14',
d15:'supplier.tbl.15',
d16:'supplier.tbl.16',
d17:'supplier.tbl.17',
d18:'supplier.tbl.18',
d19:'supplier.tbl.19',
d20:'supplier.tbl.20',
d21:'supplier.tbl.21',
d22:'supplier.tbl.22',
d23:'supplier.tbl.23',
d24:'supplier.tbl.24',
d25:'supplier.tbl.25',
d26:'supplier.tbl.26',
d27:'supplier.tbl.27',
d28:'supplier.tbl.28',
d29:'supplier.tbl.29',
d30:'supplier.tbl.30',
d31:'supplier.tbl.31',
d32:'supplier.tbl.32',
d33:'supplier.tbl.33',
d34:'supplier.tbl.34',
d35:'supplier.tbl.35',
d36:'supplier.tbl.36',
d37:'supplier.tbl.37',
d38:'supplier.tbl.38',
d39:'supplier.tbl.39',
d40:'supplier.tbl.40',
d41:'supplier.tbl.41',
d42:'supplier.tbl.42',
d43:'supplier.tbl.43',
d44:'supplier.tbl.44',
d45:'supplier.tbl.45',
d46:'supplier.tbl.46',
d47:'supplier.tbl.47',
```

```
d48:'supplier.tbl.48',
d49:'supplier.tbl.49',
d50:'supplier.tbl.50',
d51:'supplier.tbl.51',
d52:'supplier.tbl.52',
d53:'supplier.tbl.53',
d54:'supplier.tbl.54',
d55:'supplier.tbl.55',
d56:'supplier.tbl.56',
d57:'supplier.tbl.57',
d58:'supplier.tbl.58',
d59:'supplier.tbl.59',
d60:'supplier.tbl.60',
d61:'supplier.tbl.61',
d62:'supplier.tbl.62',
d63:'supplier.tbl.63',
d64:'supplier.tbl.64'
))reject limit unlimited;
alter table supplier_et parallel;

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

organization external (
type ORACLE_LOADER
default directory d1
access parameters
(
      records delimited by newline
      nobadfile
      nologfile
      nodiscardfile
      fields terminated by '|'
      missing field values are null
)
location (
d1:'customer.tbl.1',
d2:'customer.tbl.2',
d3:'customer.tbl.3',
d4:'customer.tbl.4',
d5:'customer.tbl.5',
d6:'customer.tbl.6',
d7:'customer.tbl.7',
d8:'customer.tbl.8',
d9:'customer.tbl.9',
d10:'customer.tbl.10',
d11:'customer.tbl.11',
d12:'customer.tbl.12',
d13:'customer.tbl.13',
d14:'customer.tbl.14',
d15:'customer.tbl.15',
d16:'customer.tbl.16',
d17:'customer.tbl.17',
d18:'customer.tbl.18',
d19:'customer.tbl.19',
d20:'customer.tbl.20',
d21:'customer.tbl.21',
d22:'customer.tbl.22',
d23:'customer.tbl.23',
d24:'customer.tbl.24',
d25:'customer.tbl.25',
d26:'customer.tbl.26',
d27:'customer.tbl.27',
d28:'customer.tbl.28',
d29:'customer.tbl.29',
d30:'customer.tbl.30',
d31:'customer.tbl.31',
d32:'customer.tbl.32',
d33:'customer.tbl.33',
d34:'customer.tbl.34',
d35:'customer.tbl.35',
d36:'customer.tbl.36',
d37:'customer.tbl.37',
d38:'customer.tbl.38',
d39:'customer.tbl.39',
d40:'customer.tbl.40',
d41:'customer.tbl.41',
d42:'customer.tbl.42',
d43:'customer.tbl.43',
d44:'customer.tbl.44',
d45:'customer.tbl.45',
d46:'customer.tbl.46',
d47:'customer.tbl.47',
d48:'customer.tbl.48',
d49:'customer.tbl.49',
d50:'customer.tbl.50',
d51:'customer.tbl.51',
d52:'customer.tbl.52',
d53:'customer.tbl.53',
d54:'customer.tbl.54',
d55:'customer.tbl.55',
d56:'customer.tbl.56',
d57:'customer.tbl.57',
d58:'customer.tbl.58',
d59:'customer.tbl.59',
d60:'customer.tbl.60',
d61:'customer.tbl.61',
d62:'customer.tbl.62',
d63:'customer.tbl.63',
d64:'customer.tbl.64'
))reject limit unlimited;
alter table customer_et parallel;

drop table nation_et;
create table nation_et (
   n_nationk        number ,
   n_name           char(25) ,
   n_regionkey      number ,
   n_comment        varchar(152)
)
organization external (
type ORACLE_LOADER
default directory d1
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 d1
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;

drop table lineitem;
create table lineitem(
   l_shipdate ,
   l_orderkey              NOT NULL,
   l_discount              NOT NULL,
   l_extendedprice         NOT NULL,
   l_suppkey               NOT NULL,
   l_quantity              NOT NULL,
   l_returnflag,
   l_partkey               NOT NULL,
   l_linestatus,
   l_tax                   NOT NULL,
   l_commitdate,
   l_receiptdate,
   l_shipmode,
   l_linenumber            NOT NULL,
   l_shipinstruct,
   l_comment
)
pctfree 1
pctused 99
initrans 10
storage (freelist groups 4 freelists 84)
compress
parallel
nologging
partition by range (l_shipdate)
subpartition by hash(l_partkey)
subpartitions 128
(
partition item1 values less than (to_date('1992-01-01','YYYY-MM-DD'))
tablespace l_1 ,
partition item2 values less than (to_date('1992-02-01','YYYY-MM-DD'))
tablespace l_2 ,
partition item3 values less than (to_date('1992-03-01','YYYY-MM-DD'))
tablespace l_3 ,
partition item4 values less than (to_date('1992-04-01','YYYY-MM-DD'))
tablespace l_4 ,
partition item5 values less than (to_date('1992-05-01','YYYY-MM-DD'))
tablespace l_5 ,
partition item6 values less than (to_date('1992-06-01','YYYY-MM-DD'))
tablespace l_6 ,
partition item7 values less than (to_date('1992-07-01','YYYY-MM-DD'))
tablespace l_7 ,
partition item8 values less than (to_date('1992-08-01','YYYY-MM-DD'))
tablespace l_8 ,
partition item9 values less than (to_date('1992-09-01','YYYY-MM-DD'))
tablespace l_9 ,
partition item10 values less than (to_date('1992-10-01','YYYY-MM-
DD')) tablespace l_10 ,
partition item11 values less than (to_date('1992-11-01','YYYY-MM-
DD')) tablespace l_11 ,
partition item12 values less than (to_date('1992-12-01','YYYY-MM-
DD')) tablespace l_12 ,
partition item13 values less than (to_date('1993-01-01','YYYY-MM-
DD')) tablespace l_13 ,
partition item14 values less than (to_date('1993-02-01','YYYY-MM-
DD')) tablespace l_14 ,
partition item15 values less than (to_date('1993-03-01','YYYY-MM-
DD')) tablespace l_15 ,
partition item16 values less than (to_date('1993-04-01','YYYY-MM-
DD')) tablespace l_16 ,
partition item17 values less than (to_date('1993-05-01','YYYY-MM-
DD')) tablespace l_17 ,
partition item18 values less than (to_date('1993-06-01','YYYY-MM-
DD')) tablespace l_18 ,
partition item19 values less than (to_date('1993-07-01','YYYY-MM-
DD')) tablespace l_19 ,
partition item20 values less than (to_date('1993-08-01','YYYY-MM-
DD')) tablespace l_20 ,
partition item21 values less than (to_date('1993-09-01','YYYY-MM-
DD')) tablespace l_21 ,
partition item22 values less than (to_date('1993-10-01','YYYY-MM-
DD')) tablespace l_22 ,
partition item23 values less than (to_date('1993-11-01','YYYY-MM-
DD')) tablespace l_23 ,
partition item24 values less than (to_date('1993-12-01','YYYY-MM-
DD')) tablespace l_24 ,
partition item25 values less than (to_date('1994-01-01','YYYY-MM-
DD')) tablespace l_25 ,
partition item26 values less than (to_date('1994-02-01','YYYY-MM-
DD')) tablespace l_26 ,
partition item27 values less than (to_date('1994-03-01','YYYY-MM-
DD')) tablespace l_27 ,
partition item28 values less than (to_date('1994-04-01','YYYY-MM-
DD')) tablespace l_28 ,
partition item29 values less than (to_date('1994-05-01','YYYY-MM-
DD')) tablespace l_29 ,
partition item30 values less than (to_date('1994-06-01','YYYY-MM-
DD')) tablespace l_30 ,
partition item31 values less than (to_date('1994-07-01','YYYY-MM-
DD')) tablespace l_31 ,
partition item32 values less than (to_date('1994-08-01','YYYY-MM-
DD')) tablespace l_32 ,
partition item33 values less than (to_date('1994-09-01','YYYY-MM-
DD')) tablespace l_33 ,
partition item34 values less than (to_date('1994-10-01','YYYY-MM-
DD')) tablespace l_34 ,
partition item35 values less than (to_date('1994-11-01','YYYY-MM-
DD')) tablespace l_35 ,
partition item36 values less than (to_date('1994-12-01','YYYY-MM-
DD')) tablespace l_36 ,
partition item37 values less than (to_date('1995-01-01','YYYY-MM-
DD')) tablespace l_37 ,
partition item38 values less than (to_date('1995-02-01','YYYY-MM-
DD')) tablespace l_38 ,
partition item39 values less than (to_date('1995-03-01','YYYY-MM-
DD')) tablespace l_39 ,
partition item40 values less than (to_date('1995-04-01','YYYY-MM-
DD')) tablespace l_40 ,
partition item41 values less than (to_date('1995-05-01','YYYY-MM-
DD')) tablespace l_41 ,
partition item42 values less than (to_date('1995-06-01','YYYY-MM-
DD')) tablespace l_42 ,
partition item43 values less than (to_date('1995-07-01','YYYY-MM-
DD')) tablespace l_43 ,
partition item44 values less than (to_date('1995-08-01','YYYY-MM-
DD')) tablespace l_44 ,
partition item45 values less than (to_date('1995-09-01','YYYY-MM-
DD')) tablespace l_45 ,
partition item46 values less than (to_date('1995-10-01','YYYY-MM-
DD')) tablespace l_46 ,
partition item47 values less than (to_date('1995-11-01','YYYY-MM-
DD')) tablespace l_47 ,
partition item48 values less than (to_date('1995-12-01','YYYY-MM-
DD')) tablespace l_48 ,
partition item49 values less than (to_date('1996-01-01','YYYY-MM-
DD')) tablespace l_49 ,
partition item50 values less than (to_date('1996-02-01','YYYY-MM-
DD')) tablespace l_50 ,
partition item51 values less than (to_date('1996-03-01','YYYY-MM-
DD')) tablespace l_51 ,
partition item52 values less than (to_date('1996-04-01','YYYY-MM-
DD')) tablespace l_52 ,
partition item53 values less than (to_date('1996-05-01','YYYY-MM-
DD')) tablespace l_53 ,
```

```
partition item54 values less than (to_date('1996-06-01','YYYY-MM-
DD')) tablespace l_54 ,
partition item55 values less than (to_date('1996-07-01','YYYY-MM-
DD')) tablespace l_55 ,
partition item56 values less than (to_date('1996-08-01','YYYY-MM-
DD')) tablespace l_56 ,
partition item57 values less than (to_date('1996-09-01','YYYY-MM-
DD')) tablespace l_57 ,
partition item58 values less than (to_date('1996-10-01','YYYY-MM-
DD')) tablespace l_58 ,
partition item59 values less than (to_date('1996-11-01','YYYY-MM-
DD')) tablespace l_59 ,
partition item60 values less than (to_date('1996-12-01','YYYY-MM-
DD')) tablespace l_60 ,
partition item61 values less than (to_date('1997-01-01','YYYY-MM-
DD')) tablespace l_61 ,
partition item62 values less than (to_date('1997-02-01','YYYY-MM-
DD')) tablespace l_62 ,
partition item63 values less than (to_date('1997-03-01','YYYY-MM-
DD')) tablespace l_63 ,
partition item64 values less than (to_date('1997-04-01','YYYY-MM-
DD')) tablespace l_64 ,
partition item65 values less than (to_date('1997-05-01','YYYY-MM-
DD')) tablespace l_65 ,
partition item66 values less than (to_date('1997-06-01','YYYY-MM-
DD')) tablespace l_66 ,
partition item67 values less than (to_date('1997-07-01','YYYY-MM-
DD')) tablespace l_67 ,
partition item68 values less than (to_date('1997-08-01','YYYY-MM-
DD')) tablespace l_68 ,
partition item69 values less than (to_date('1997-09-01','YYYY-MM-
DD')) tablespace l_69 ,
partition item70 values less than (to_date('1997-10-01','YYYY-MM-
DD')) tablespace l_70 ,
partition item71 values less than (to_date('1997-11-01','YYYY-MM-
DD')) tablespace l_71 ,
partition item72 values less than (to_date('1997-12-01','YYYY-MM-
DD')) tablespace l_72 ,
partition item73 values less than (to_date('1998-01-01','YYYY-MM-
DD')) tablespace l_73 ,
partition item74 values less than (to_date('1998-02-01','YYYY-MM-
DD')) tablespace l_74 ,
partition item75 values less than (to_date('1998-03-01','YYYY-MM-
DD')) tablespace l_75 ,
partition item76 values less than (to_date('1998-04-01','YYYY-MM-
DD')) tablespace l_76 ,
partition item77 values less than (to_date('1998-05-01','YYYY-MM-
DD')) tablespace l_77 ,
partition item78 values less than (to_date('1998-06-01','YYYY-MM-
DD')) tablespace l_78 ,
partition item79 values less than (to_date('1998-07-01','YYYY-MM-
DD')) tablespace l_79 ,
partition item80 values less than (to_date('1998-08-01','YYYY-MM-
DD')) tablespace l_80 ,
partition item81 values less than (to_date('1998-09-01','YYYY-MM-
DD')) tablespace l_81 ,
partition item82 values less than (to_date('1998-10-01','YYYY-MM-
DD')) tablespace l_82 ,
partition item83 values less than (to_date('1998-11-01','YYYY-MM-
DD')) tablespace l_83 ,
partition item84 values less than (MAXVALUE) tablespace l_84
)
as select
    l_shipdate,
    l_orderkey,
    l_discount,
    l_extendedprice,
    l_suppkey,
    l_quantity,
    l_returnflag,
    l_partkey,
    l_linestatus,
    l_tax,
    l_commitdate,
    l_receiptdate,
    l_shipmode,
    l_linenumber,
    l_shipinstruct,
    l_comment
from lineitem_et;
drop table 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
storage (freelist groups 4 freelists 84)
compress
parallel
nologging
partition by range (o_orderdate)
subpartition by hash(o_custkey)
subpartitions 128
(
partition ord1 values less than (to_date('1992-01-01','YYYY-MM-DD'))
tablespace o_1 ,
partition ord2 values less than (to_date('1992-02-01','YYYY-MM-DD'))
tablespace o_2 ,
partition ord3 values less than (to_date('1992-03-01','YYYY-MM-DD'))
tablespace o_3 ,
partition ord4 values less than (to_date('1992-04-01','YYYY-MM-DD'))
tablespace o_4 ,
partition ord5 values less than (to_date('1992-05-01','YYYY-MM-DD'))
tablespace o_5 ,
partition ord6 values less than (to_date('1992-06-01','YYYY-MM-DD'))
tablespace o_6 ,
partition ord7 values less than (to_date('1992-07-01','YYYY-MM-DD'))
tablespace o_7 ,
partition ord8 values less than (to_date('1992-08-01','YYYY-MM-DD'))
tablespace o_8 ,
partition ord9 values less than (to_date('1992-09-01','YYYY-MM-DD'))
tablespace o_9 ,
partition ord10 values less than (to_date('1992-10-01','YYYY-MM-DD'))
tablespace o_10 ,
partition ord11 values less than (to_date('1992-11-01','YYYY-MM-DD'))
tablespace o_11 ,
partition ord12 values less than (to_date('1992-12-01','YYYY-MM-DD'))
tablespace o_12 ,
partition ord13 values less than (to_date('1993-01-01','YYYY-MM-DD'))
tablespace o_13 ,
partition ord14 values less than (to_date('1993-02-01','YYYY-MM-DD'))
tablespace o_14 ,
partition ord15 values less than (to_date('1993-03-01','YYYY-MM-DD'))
tablespace o_15 ,
partition ord16 values less than (to_date('1993-04-01','YYYY-MM-DD'))
tablespace o_16 ,
partition ord17 values less than (to_date('1993-05-01','YYYY-MM-DD'))
tablespace o_17 ,
partition ord18 values less than (to_date('1993-06-01','YYYY-MM-DD'))
tablespace o_18 ,
partition ord19 values less than (to_date('1993-07-01','YYYY-MM-DD'))
tablespace o_19 ,
partition ord20 values less than (to_date('1993-08-01','YYYY-MM-DD'))
tablespace o_20 ,
partition ord21 values less than (to_date('1993-09-01','YYYY-MM-DD'))
tablespace o_21 ,
```

```
partition ord22 values less than (to_date('1993-10-01','YYYY-MM-DD'))
tablespace o_22 ,
partition ord23 values less than (to_date('1993-11-01','YYYY-MM-DD'))
tablespace o_23 ,
partition ord24 values less than (to_date('1993-12-01','YYYY-MM-DD'))
tablespace o_24 ,
partition ord25 values less than (to_date('1994-01-01','YYYY-MM-DD'))
tablespace o_25 ,
partition ord26 values less than (to_date('1994-02-01','YYYY-MM-DD'))
tablespace o_26 ,
partition ord27 values less than (to_date('1994-03-01','YYYY-MM-DD'))
tablespace o_27 ,
partition ord28 values less than (to_date('1994-04-01','YYYY-MM-DD'))
tablespace o_28 ,
partition ord29 values less than (to_date('1994-05-01','YYYY-MM-DD'))
tablespace o_29 ,
partition ord30 values less than (to_date('1994-06-01','YYYY-MM-DD'))
tablespace o_30 ,
partition ord31 values less than (to_date('1994-07-01','YYYY-MM-DD'))
tablespace o_31 ,
partition ord32 values less than (to_date('1994-08-01','YYYY-MM-DD'))
tablespace o_32 ,
partition ord33 values less than (to_date('1994-09-01','YYYY-MM-DD'))
tablespace o_33 ,
partition ord34 values less than (to_date('1994-10-01','YYYY-MM-DD'))
tablespace o_34 ,
partition ord35 values less than (to_date('1994-11-01','YYYY-MM-DD'))
tablespace o_35 ,
partition ord36 values less than (to_date('1994-12-01','YYYY-MM-DD'))
tablespace o_36 ,
partition ord37 values less than (to_date('1995-01-01','YYYY-MM-DD'))
tablespace o_37 ,
partition ord38 values less than (to_date('1995-02-01','YYYY-MM-DD'))
tablespace o_38 ,
partition ord39 values less than (to_date('1995-03-01','YYYY-MM-DD'))
tablespace o_39 ,
partition ord40 values less than (to_date('1995-04-01','YYYY-MM-DD'))
tablespace o_40 ,
partition ord41 values less than (to_date('1995-05-01','YYYY-MM-DD'))
tablespace o_41 ,
partition ord42 values less than (to_date('1995-06-01','YYYY-MM-DD'))
tablespace o_42 ,
partition ord43 values less than (to_date('1995-07-01','YYYY-MM-DD'))
tablespace o_43 ,
partition ord44 values less than (to_date('1995-08-01','YYYY-MM-DD'))
tablespace o_44 ,
partition ord45 values less than (to_date('1995-09-01','YYYY-MM-DD'))
tablespace o_45 ,
partition ord46 values less than (to_date('1995-10-01','YYYY-MM-DD'))
tablespace o_46 ,
partition ord47 values less than (to_date('1995-11-01','YYYY-MM-DD'))
tablespace o_47 ,
partition ord48 values less than (to_date('1995-12-01','YYYY-MM-DD'))
tablespace o_48 ,
partition ord49 values less than (to_date('1996-01-01','YYYY-MM-DD'))
tablespace o_49 ,
partition ord50 values less than (to_date('1996-02-01','YYYY-MM-DD'))
tablespace o_50 ,
partition ord51 values less than (to_date('1996-03-01','YYYY-MM-DD'))
tablespace o_51 ,
partition ord52 values less than (to_date('1996-04-01','YYYY-MM-DD'))
tablespace o_52 ,
partition ord53 values less than (to_date('1996-05-01','YYYY-MM-DD'))
tablespace o_53 ,
partition ord54 values less than (to_date('1996-06-01','YYYY-MM-DD'))
tablespace o_54 ,
partition ord55 values less than (to_date('1996-07-01','YYYY-MM-DD'))
tablespace o_55 ,
partition ord56 values less than (to_date('1996-08-01','YYYY-MM-DD'))
tablespace o_56 ,
partition ord57 values less than (to_date('1996-09-01','YYYY-MM-DD'))
tablespace o_57 ,
partition ord58 values less than (to_date('1996-10-01','YYYY-MM-DD'))
tablespace o_58 ,
partition ord59 values less than (to_date('1996-11-01','YYYY-MM-DD'))
tablespace o_59 ,
partition ord60 values less than (to_date('1996-12-01','YYYY-MM-DD'))
tablespace o_60 ,
partition ord61 values less than (to_date('1997-01-01','YYYY-MM-DD'))
tablespace o_61 ,
partition ord62 values less than (to_date('1997-02-01','YYYY-MM-DD'))
tablespace o_62 ,
partition ord63 values less than (to_date('1997-03-01','YYYY-MM-DD'))
tablespace o_63 ,
partition ord64 values less than (to_date('1997-04-01','YYYY-MM-DD'))
tablespace o_64 ,
partition ord65 values less than (to_date('1997-05-01','YYYY-MM-DD'))
tablespace o_65 ,
partition ord66 values less than (to_date('1997-06-01','YYYY-MM-DD'))
tablespace o_66 ,
partition ord67 values less than (to_date('1997-07-01','YYYY-MM-DD'))
tablespace o_67 ,
partition ord68 values less than (to_date('1997-08-01','YYYY-MM-DD'))
tablespace o_68 ,
partition ord69 values less than (to_date('1997-09-01','YYYY-MM-DD'))
tablespace o_69 ,
partition ord70 values less than (to_date('1997-10-01','YYYY-MM-DD'))
tablespace o_70 ,
partition ord71 values less than (to_date('1997-11-01','YYYY-MM-DD'))
tablespace o_71 ,
partition ord72 values less than (to_date('1997-12-01','YYYY-MM-DD'))
tablespace o_72 ,
partition ord73 values less than (to_date('1998-01-01','YYYY-MM-DD'))
tablespace o_73 ,
partition ord74 values less than (to_date('1998-02-01','YYYY-MM-DD'))
tablespace o_74 ,
partition ord75 values less than (to_date('1998-03-01','YYYY-MM-DD'))
tablespace o_75 ,
partition ord76 values less than (to_date('1998-04-01','YYYY-MM-DD'))
tablespace o_76 ,
partition ord77 values less than (to_date('1998-05-01','YYYY-MM-DD'))
tablespace o_77 ,
partition ord78 values less than (to_date('1998-06-01','YYYY-MM-DD'))
tablespace o_78 ,
partition ord79 values less than (to_date('1998-07-01','YYYY-MM-DD'))
tablespace o_79 ,
partition ord80 values less than (to_date('1998-08-01','YYYY-MM-DD'))
tablespace o_80 ,
partition ord81 values less than (to_date('1998-09-01','YYYY-MM-DD'))
tablespace o_81 ,
partition ord82 values less than (to_date('1998-10-01','YYYY-MM-DD'))
tablespace o_82 ,
partition ord83 values less than (to_date('1998-11-01','YYYY-MM-DD'))
tablespace o_83 ,
partition ord84 values less than (MAXVALUE) tablespace o_84

)
as select
   o_orderdate,
   o_orderkey,
   o_custkey,
   o_orderpriority,
   o_shippriority,
   o_clerk,
   o_orderstatus,
   o_totalprice,
   o_comment
from orders_et;
drop table orders_et;

drop table partsupp;
create table partsupp(
   ps_partkey          NOT NULL ,
   ps_suppkey          NOT NULL ,
   ps_supplycost       NOT NULL ,
   ps_availqty,
   ps_comment
```

```
   , constraint pk_partkey_suppkey_1 primary
key(ps_partkey,ps_suppkey)
)
organization index
pctthreshold 50
storage (initial 720M)
tablespace ts_ps
compress
parallel
nologging
partition by hash (ps_partkey)
partitions 128
as select
   ps_partkey,
   ps_suppkey,
   ps_supplycost,
   ps_availqty,
   ps_comment
from partsupp_et;
drop table 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 0
pctused 99
storage (initial 720M)
tablespace ts_p
parallel
nologging
partition by hash (p_partkey)
partitions 128
as select
   p_partkey,
   p_type,
   p_size,
   p_brand,
   p_name,
   p_container,
   p_mfgr,
   p_retailprice,
   p_comment
from part_et;
drop table 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 0
pctused 99
storage (initial 720M )
tablespace ts_c
parallel
nologging
partition by hash (c_custkey)
partitions 128
as select
   c_custkey,
   c_mktsegment ,
   c_nationkey,
   c_name,
   c_address,
   c_phone,
   c_acctbal,
   c_comment
from customer_et;
drop table 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 0
pctused 99
storage (initial  60M)
tablespace ts_s
parallel
nologging
partition by hash (s_suppkey)
partitions 128
as select
   s_suppkey,
   s_nationkey,
   s_comment,
   s_name,
   s_address,
   s_phone,
   s_acctbal
from supplier_et;
drop table supplier_et;

drop table nation;
create table nation(
   n_nationkey        NOT NULL ,
   n_name,
   n_regionkey,
   n_comment
)
tablespace ts_default
as select * from nation_et;
drop table nation_et;

drop table region;
create table region(
   r_regionkey,
   r_name,
   r_comment
)
tablespace ts_default
as select * from region_et;
drop table region_et;
!

echo End Table Creation at `date`
```

B.6 anlyz_3k.sh

```
#!/bin/ksh

echo Start Analyze at `date`

sqlplus tpch/tpch <<!
set echo on
set timing on
```

```
set termout on
execute dbms_stats.gather_schema_stats('TPCH' , estimate_percent
=> 1, degree => 128 , granularity => 'GLOBAL' );
exit;
!

sqlplus tpch/tpch <<!
connect / as sysdba
execute dbms_stats.gather_system_stats;
exec dbms_scheduler.disable('GATHER_STATS_JOB');
exit;
!


echo End Analyze at `date`
```

Appendix - C Query Text and Output

C.1 Qualification Query Output

Query 1

```
-- TPC-H/TPC-R Pricing Summary Report Query (Q1)

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            3.7734107E+007 5.6586554400730003E+010
5.3758257134869987E+010 5.5909065222827721E+010
2.5522005853257337E+001
3.8273129734621674E+004 4.9985295838395505E-002 1478493.00
N            F            9.91417E+005 1.4875047103800001E+009
1.4130821680540993E+009 1.4696492231943748E+009
2.5516471920522985E+001
3.8284467760848303E+004 5.009342667421629E-002 38854.00
N            O            7.447604E+007 1.1170172969773997E+011
1.0611823030760562E+011 1.1036704387249698E+011
2.5502226769584 99E+001
3.8249117988908263E+004 4.9996586053695606E-002 2920374.00
R            F            3.7719753E+007 5.6568041380900055E+010
5.3741292684603989E+010 5.5889619119831978E+010
2.5505793612690 77E+001
3.8250854626099695E+004 5.0009405830124976E-002 1478870.00


4 rows processed.
```

Query 2

```
-- TPC-H/TPC-R Minimum Cost Supplier Query (Q2)


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
9.9385300000000007E+003 Supplier#000005359        UNITED
KINGDOM
185358.00           Manufacturer#4
QKuHYh,vZGiwu2FWEJoLDx04               33-429-790-6131
blithely silent pinto beans are furiously. slyly final deposits acros
9.9378400000000001E+003 Supplier#000005969        ROMANIA
108438.00           Manufacturer#1
ANDENSOSmk,miq23Xfb5RWt6dvUcvt6Qa        29-520-692-3537
carefully slow deposits use furiously. slyly ironic platelets above the
ironic
9.9362199999999993E+003 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
9.9237700000000004E+003 Supplier#000002324        GERMANY
29821.00            Manufacturer#4
y3OD9UywSTOk                           17-779-299-1839
quickly express packages breach quiet pinto beans. requ
9.8712199999999993E+003 Supplier#000006373        GERMANY
43868.00            Manufacturer#5
J8fcXWsTqM                             17-813-485-8637
never silent deposits integrate furiously blit
9.8707800000000007E+003 Supplier#000001286        GERMANY
81285.00            Manufacturer#2
YKA,E2fjiVd7eUrzp2Ef8j1QxGo2DFnosaTEH    17-516-924-4574
final theodolites cajole slyly special,
9.8707800000000007E+003 Supplier#000001286        GERMANY
181285.00           Manufacturer#4
YKA,E2fjiVd7eUrzp2Ef8j1QxGo2DFnosaTEH    17-516-924-4574
```

final theodolites cajole slyly special,
9.8525200000000004E+003 Supplier#000008973 RUSSIA
18972.00 Manufacturer#2
t5L67YdBYYH6o,Vz24jpDyQ9 32-188-594-7038
quickly regular instructions wake-- carefully unusual braids into the expres
9.8478299999999999E+003 Supplier#000008097 RUSSIA
130557.00 Manufacturer#2
xMe97bpE69NzdwLoX 32-375-640-3593
slyly regular dependencies sleep slyly furiously express dep
9.8475699999999997E+003 Supplier#000006345 FRANCE
86344.00 Manufacturer#1
VSt3rzk3qG698u6ld8HhOByvrTcSTSvQlDQDag 16-886-766-7945
silent pinto beans should have to snooze carefully along the final reques
9.8475699999999997E+003 Supplier#000006345 FRANCE
173827.00 Manufacturer#2
VSt3rzk3qG698u6ld8HhOByvrTcSTSvQlDQDag 16-886-766-7945
silent pinto beans should have to snooze carefully along the final reques
9.8369300000000003E+003 Supplier#000007342 RUSSIA
4841.00 Manufacturer#4
JOlK7C1,7xrEZSSOw 32-399-414-5385
final accounts haggle. bold accounts are furiously dugouts. furiously silent asymptotes are slyly
9.8171000000000004E+003 Supplier#000002352 RUSSIA
124815.00 Manufacturer#2
4LfoHUZjgjEbAKw TgdKcgOc4D4uCYw 32-551-831-1437
blithely pending packages across the ironic accounts grow slyly after the furiously

------ rows deleted ------

7.9806499999999996E+003 Supplier#000001288 FRANCE
13784.00 Manufacturer#4
zE,7HgVPrCn 16-646-464-8247
unusual pinto beans cajole furiously according t
7.9503699999999999E+003 Supplier#000008101 GERMANY
33094.00 Manufacturer#5
kkYvL6IuvojJgTNG IKkaXQDYgx8ILohj 17-627-663-8014
quickly regular requests are furiously. pending deposits wake
7.9379300000000003E+003 Supplier#000009012 ROMANIA
83995.00 Manufacturer#2
iUiTziH,Ek3i4lwSgunXMgrcTzwdb 29-250-925-9690
blithely bold ideas haggle quickly final, regular request
7.9144499999999998E+003 Supplier#000001013 RUSSIA
125988.00 Manufacturer#2
riRcntps4KEDtYScjpMIWeYF6mNnR 32-194-698-3365
final, ironic theodolites alongside of the ironic
7.9129099999999999E+003 Supplier#000004211 GERMANY
159180.00 Manufacturer#5
2wQRVovHrm3,v03IKzfTd,1PYsFXQFFOG 17-266-947-7315
final requests integrate slyly above the silent, even
7.9129099999999999E+003 Supplier#000004211 GERMANY
184210.00 Manufacturer#4
2wQRVovHrm3,v03IKzfTd,1PYsFXQFFOG 17-266-947-7315
final requests integrate slyly above the silent, even
7.8945600000000004E+003 Supplier#000007981 GERMANY
85472.00 Manufacturer#4
NSJ96vMROAbeXP 17-963-404-3760
regular, even theodolites integrate carefully. bold, special theodolites are slyly fluffily iron
7.8870799999999999E+003 Supplier#000009792 GERMANY
164759.00 Manufacturer#3
Y28ITVeYriT3kIGdV2K8fSZ V2UqT5H1Otz 17-988-938-4296
pending, ironic packages sleep among the carefully ironic accounts. quickly final accounts
7.8715E+003 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
7.8524499999999998E+003 Supplier#000005864 RUSSIA
8363.00 Manufacturer#4
WCNfBPZeSXh3h,c 32-454-883-3821
blithely regular deposits
7.8506599999999999E+003 Supplier#000001518 UNITED KINGDOM
86501.00 Manufacturer#1
ONda3YJiHKJOC 33-730-383-3892
furiously final accounts wake carefully idle requests. even dolphins wake acc
7.8435200000000004E+003 Supplier#000006683 FRANCE
11680.00 Manufacturer#4
2Z0JGkiv01Y00oCFwUGfvilbhzCdy 16-464-517-8943
carefully bold accounts doub

100 rows processed.

Query 3

```
-- TPC-H/TPC-R Shipping Priority Query (Q3)

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          4.061810111E+005 1995-03-05  0.00
3459808.00          4.0583869889999996E+005 1995-03-04  0.00
492164.00           3.9032406099999999E+005 1995-02-19  0.00
1188320.00          3.8453793590000004E+005 1995-03-09  0.00
2435712.00          3.7867305579999997E+005 1995-02-26  0.00
4878020.00          3.7837679519999993E+005 1995-03-12  0.00
5521732.00          3.7515392150000005E+005 1995-03-13  0.00
2628192.00          3.7313330940000003E+005 1995-02-22  0.00
993600.00           3.7140745950000006E+005 1995-03-05  0.00
2300070.00          3.6737114520000003E+005 1995-03-13  0.00
```

10 rows processed.

Query 4

```
-- TPC-H/TPC-R Order Priority Checking Query (Q4)

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 5

```
-- TPC-H/TPC-R Local Supplier Volume Query (Q5)

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                  5.5502041169699997E+007
VIETNAM                   5.5295086996699966E+007
CHINA                     5.372449425659997E+007
INDIA                     5.2035512000200011E+007
JAPAN                      4.5410175695399992E+007


5 rows processed.
```

Query 6

```
-- TPC-H/TPC-R Forecasting Revenue Change Query (Q6)

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
1.231410782282999E+008


1 row processed.
```

Query 7

```
-- TPC-H/TPC-R Volume Shipping Query (Q7)

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
5.4639732733599991E+007
FRANCE               GERMANY              1996.00
5.4633083307599984E+007
GERMANY              FRANCE               1995.00
5.2531746669700004E+007
GERMANY              FRANCE               1996.00
5.2520549022400022E+007
```

4 rows processed.

Query 8

```
-- TPC-H/TPC-R National Market Share Query (Q8) -- Variant A

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	3.4435890406654783E-002
1996.00	4.1485521293530316E-002

2 rows processed.

Query 9

```
-- TPC-H/TPC-R Product Type Profit Measure Query (Q9)

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

NATION	O_YEAR	SUM_PROFIT
ALGERIA	1998.00	3.134286723450001E+007
ALGERIA	1997.00	5.7138193023299977E+007
ALGERIA	1996.00	5.6140140132999994E+007
ALGERIA	1995.00	5.3051469653399982E+007
ALGERIA	1994.00	5.3867582128600016E+007
ALGERIA	1993.00	5.4942718132399976E+007
ALGERIA	1992.00	5.4628034712699987E+007
ARGENTINA	1998.00	3.0211185708099995E+007
ARGENTINA	1997.00	5.0805741752299979E+007
ARGENTINA	1996.00	5.1923746575500026E+007
ARGENTINA	1995.00	4.929862576659999E+007
ARGENTINA	1994.00	5.0835610109500021E+007
ARGENTINA	1993.00	5.1646079177499987E+007
ARGENTINA	1992.00	5.0410314994799957E+007
BRAZIL	1998.00	2.721792438320002E+007
BRAZIL	1997.00	4.8378669198900007E+007
BRAZIL	1996.00	5.0482870357199967E+007
BRAZIL	1995.00	4.7623383634900019E+007
BRAZIL	1994.00	4.7840165725600019E+007
BRAZIL	1993.00	4.905469403510002E+007
BRAZIL	1992.00	4.8667639084200002E+007
CANADA	1998.00	3.0379833768500008E+007
CANADA	1997.00	5.0465052311400011E+007
CANADA	1996.00	5.2560501390399992E+007
CANADA	1995.00	5.2375332809200019E+007
CANADA	1994.00	5.2600364658699982E+007
CANADA	1993.00	5.2644504073499992E+007
CANADA	1992.00	5.3932871696999989E+007
CHINA	1998.00	3.1075466164899986E+007
CHINA	1997.00	5.0551874449900024E+007
CHINA	1996.00	5.1039293875399999E+007
CHINA	1995.00	4.9287534616900027E+007
CHINA	1994.00	5.0851090067399979E+007
CHINA	1993.00	5.4229629833000019E+007
CHINA	1992.00	5.2400529371999994E+007
EGYPT	1998.00	2.9054433385600001E+007
EGYPT	1997.00	5.0627611452400021E+007
EGYPT	1996.00	4.9542212844599977E+007
EGYPT	1995.00	4.831155032069999E+007
EGYPT	1994.00	4.9790644736000001E+007
EGYPT	1993.00	

------ rows deleted ------

5.0077306418599971E+007
UNITED STATES 1996.00
4.8048649470300019E+007
UNITED STATES 1995.00
4.8809032422600001E+007
UNITED STATES 1994.00
4.9296747182700016E+007
UNITED STATES 1993.00
4.8029946801400028E+007
UNITED STATES 1992.00
4.8671944498299979E+007

```
VIETNAM         1998.00         3.0442736059399996E+007
VIETNAM         1997.00         5.0309179794200003E+007
VIETNAM         1996.00         5.0488161409999967E+007
VIETNAM         1995.00         4.9658284612500004E+007
VIETNAM         1994.00         5.0596057260699995E+007
VIETNAM         1993.00         5.0953919151900016E+007
VIETNAM         1992.00         4.9613838315100007E+007


175 rows processed.
```

Query 10

```
-- TPC-H/TPC-R Returned Item Reporting Query (Q10)

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
orders,
customer,
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#0000057040
7.3423524550000008E+005 6.3287E+002
JAPAN           Eioyzjf4pp                      22-895-641-3466
requests sleep blithely about the furiously i
143347.00       Customer#0000143347
7.2100269480000006E+005 2.5574699999999998E+003
EGYPT           1aReFYv,Kw4                     14-742-935-
3718
fluffily bold excuses haggle finally after the u
60838.00        Customer#0000060838
6.7912730769999989E+005 2.45477E+003
BRAZIL          64EaJ5vMAHWJlBOxJklpNc2RJiWE    12-
913-494-9813
furiously even pinto beans integrate under the ruthless foxes; ironic,
even dolphins across the slyl
101998.00       Customer#0000101998
6.3702956669999997E+005 3.7908899999999999E+003
UNITED KINGDOM  01c9CILnNtfOQYmZj               33-
593-865-6378
accounts doze blithely! enticing, final deposits sleep blithely special
accounts. slyly express accounts pla
125341.00       Customer#0000125341
6.3350808599999989E+005 4.9835100000000002E+003
GERMANY         S29ODD6bceU8QSuuEJznkNaK        17-
582-695-5962
quickly express requests wake quickly blithely
25501.00        Customer#0000025501
6.2026978490000009E+005 7.72504E+003
ETHIOPIA        W556MXuoiaYCCZamJI,Rn0B4ACUGdkQ8DZ
15-874-808-6793
quickly special requests sleep evenly among the special deposits.
special deposi
115831.00       Customer#0000115831
5.964238672000001E+005 5.0981000000000004E+003
FRANCE          rFeBbEEyk dl ne7zV5fDrmiq1oK09wV7pxqCgIc
16-715-386-3788
carefully bold excuses sleep alongside of the thinly idle
84223.00        Customer#0000084223
5.9499802390000003E+005 5.2864999999999998E+002
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#0000054289
5.856033918000001E+005 5.5830200000000004E+003
IRAN            vXCxoCsU0Bad5JQI ,oobkZ         20-834-
292-4707
express requests sublate blithely regular requests. regular, even ideas
solve.
39922.00        Customer#0000039922
5.8487811340000003E+005 7.3211099999999997E+003
GERMANY         Zgy4s50l2GKN4pLDPBU8m342glw6R
17-147-757-8036
even pinto beans haggle. slyly bold accounts inte
6226.00         Customer#0000006226
5.7678376059999992E+005 2.2300900000000001E+003
UNITED KINGDOM  8gPu8,NPGkfyQQ0hclYUGPIBWc,ybP5g,
33-657-701-3391
quickly final requests against the regular instructions wake blithely final
instructions. pa
922.00          Customer#0000000922
5.7676753330000001E+005 3.86925E+003
GERMANY         Az9RFaut7NkPnc5zSD2PwHgVwr4jRzq
17-945-916-9648
boldly final requests cajole blith
147946.00       Customer#0000147946
5.7645513199999998E+005 2.0301300000000001E+003
ALGERIA         iANyZHjqhyy7Ajah0pTrYyhJ        10-886-
956-3143
furiously even accounts are blithely above the furiousl
115640.00       Customer#0000115640
5.6934119329999981E+005 6.4361000000000004E+003
ARGENTINA       Vtgfia9qI 7EpHgecU1X            11-411-
543-4901
final instructions are slyly according to the
73606.00        Customer#0000073606     5.686568578E+005
1.7856700000000001E+003
JAPAN           xuR0Tro5yChDfOCrjkd2ol          22-437-653-
6966
furiously bold orbits about the furiously busy requests wake across the
furiously quiet theodolites. d
110246.00       Customer#0000110246
5.6684298149999999E+005 7.7633500000000004E+003
VIETNAM         7KzflgX MDOq7sOkI               31-943-426-
9837
dolphins sleep blithely among the slyly final
142549.00       Customer#0000142549
5.6353723680000007E+005 5.0859899999999998E+003
INDONESIA       ChqEoK43OysjdHbtKCp6dKqjNyvvi9  19-
955-562-2398
regular, unusual dependencies boost slyly; ironic attainments nag
fluffily into the unusual packages?
```

146149.00 Customer#0000146149 5.572549865E+005
1.79155E+003
ROMANIA s87fvzFQpU 29-744-164-
6487
silent, unusual requests detect quickly slyly regul
52528.00 Customer#0000052528
5.5639735089999996E+005 5.5178999999999996E+002
ARGENTINA NFztyTOR10UOJ 11-208-
192-3205
unusual requests detect. slyly dogged theodolites use slyly. deposit
23431.00 Customer#0000023431
5.5426953600000008E+005 3.3818600000000001E+003
ROMANIA HgiV0phqhaIa9aydNoIlb 29-915-
458-2654
instructions nag quickly. furiously bold accounts cajol

20 rows processed.

Query 11

```
-- TPC-H/TPC-R Important Stock Identification Query (Q11)

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	1.7538456859999999E+007
166726.00	1.650335392E+007
191287.00	1.6474801969999999E+007
161758.00	1.6101755539999999E+007
34452.00	1.5983844720000001E+007
139035.00	1.590707834E+007
9403.00	1.5451755620000001E+007
154358.00	1.5212937879999999E+007
38823.00	1.5064802859999999E+007
85606.00	1.5053957150000002E+007
33354.00	1.44082974E+007
154747.00	1.440758068E+007
82865.00	1.4235489780000001E+007
76094.00	1.4094247039999999E+007
222.00	1.393777774E+007
121271.00	1.3908336E+007
55221.00	1.3716120470000001E+007
22819.00	1.3666434280000001E+007
76281.00	1.364685368E+007
85298.00	1.358115493E+007
85158.00	1.3554904E+007
139684.00	1.3535538719999999E+007
31034.00	1.349802525E+007
87305.00	1.3482847039999999E+007
10181.00	1.344514875E+007
62323.00	1.3411824300000001E+007
26489.00	1.3377256380000001E+007
96493.00	1.333905783E+007
56548.00	1.3329014969999999E+007
55576.00	1.3306843350000001E+007
159751.00	1.330661448E+007
92406.00	1.32874145E+007
182636.00	1.3223726739999998E+007
199969.00	1.3135288210000001E+007
62865.00	1.3001926940000001E+007
7284.00	1.2945298190000001E+007
197867.00	1.294451052E+007
11562.00	1.293157551E+007
75165.00	1.2916918120000001E+007
97175.00	1.29112835E+007
140840.00	1.289656223E+007
65241.00	1.2890600460000001E+007
166120.00	1.2876927220000001E+007
9035.00	1.2863828699999999E+007
144616.00	1.2853549299999999E+007
176723.00	1.283230974E+007
170884.00	1.2792136579999998E+007
29790.00	1.272330033E+007
95213.00	1.255548373E+007
183873.00	1.2550533050000001E+007
171235.00	1.2476538300000001E+007
21533.00	1.243782132E+007
17290.00	1.24321595E+007
156397.00	1.22606235E+007
122611.00	1.222281298E+007
139155.00	1.222031925E+007
146316.00	1.2215800609999999E+007
171381.00	1.219973452E+007
198633.00	1.2078226949999999E+007
167417.00	1.2046637620000001E+007
59512.00	1.204346876E+007
31688.00	1.2034893640000001E+007
159586.00	1.2001505840000002E+007
8993.00	1.1963814300000001E+007
120302.00	1.1857707549999999E+007
43536.00	1.177934052E+007
9552.00	1.177690916E+007
86223.00	1.177220508E+007
53776.00	1.175866965E+007
131285.00	1.161695374E+007
91628.00	1.161111483E+007
169644.00	1.1567959719999999E+007
182299.00	1.1567462050000001E+007
33107.00	1.145381876E+007
104184.00	1.1436657440000001E+007
67027.00	1.1419127140000001E+007
176869.00	1.1371451709999999E+007
30885.00	1.1369674789999999E+007
54420.00	1.1345076879999999E+007
72240.00	1.1313951049999999E+007
178708.00	1.1294635169999998E+007
81298.00	1.1273686130000001E+007
158324.00	1.1243442720000001E+007
117095.00	1.124253524E+007
176793.00	1.1237733379999999E+007
86091.00	1.1177793789999999E+007
116033.00	1.1145434359999999E+007

------ rows deleted ------

56137.00	7.9714054000000004E+006
64729.00	7.9707697200000007E+006
98643.00	7.9686037299999995E+006
153787.00	7.9675355800000001E+006

```
8932.00          7.9672221899999995E+006
20134.00         7.9657132800000003E+006
197635.00        7.9635075800000001E+006
80408.00         7.9633121699999999E+006
37728.00         7.9618756799999997E+006
26624.00         7.9617723100000005E+006
44736.00         7.9611440999999996E+006
29763.00         7.9606050300000003E+006
36147.00         7.9594636799999997E+006
146040.00        7.9575876600000001E+006
115469.00        7.9574851400000006E+006
142276.00        7.9567906299999999E+006
181280.00        7.9540373499999996E+006
115096.00        7.9530475499999998E+006
109650.00        7.9522587300000004E+006
93862.00         7.9519922400000002E+006
158325.00        7.9507282999999998E+006
55952.00         7.9503870600000005E+006
122397.00        7.9471062700000005E+006
28114.00         7.9469457200000007E+006
11966.00         7.9451974800000004E+006
47814.00         7.944083E+006
85096.00         7.9436910599999996E+006
51657.00         7.9435937700000005E+006
196680.00        7.9435788899999997E+006
13141.00         7.9427303400000008E+006
193327.00        7.94103625E+006
152612.00        7.94066371E+006
139680.00        7.9392423599999994E+006
31134.00         7.9383182999999998E+006
45636.00         7.9372408499999996E+006
56694.00         7.9360159499999993E+006
8114.00          7.9339218799999999E+006
71518.00         7.9322616900000004E+006
72922.00         7.9304006399999997E+006
146699.00        7.9291674000000004E+006
92387.00         7.9289726699999999E+006
186289.00        7.9287861899999995E+006
95952.00         7.9279727800000003E+006
196514.00        7.9271807000000002E+006
4403.00          7.925729040000001E+006
2267.00          7.9256493700000001E+006
45924.00         7.9250476799999997E+006
11493.00         7.9167222300000004E+006
104478.00        7.9162536000000006E+006
166794.00        7.913842E+006
161995.00        7.9108742699999996E+006
23538.00         7.9097520600000005E+006
41093.00         7.9095799199999999E+006
112073.00        7.9086175700000003E+006
92814.00         7.9082625E+006
88919.00         7.9079925E+006
79753.00         7.9079338799999999E+006
108765.00        7.9053389800000004E+006
146530.00        7.9053365999999996E+006
71475.00         7.9033675800000001E+006
36289.00         7.9019465E+006
61739.00         7.900794E+006
52338.00         7.8986380800000001E+006
194299.00        7.8984212400000002E+006
105235.00        7.8978299399999995E+006
77207.00         7.8977527199999997E+006
96712.00         7.8975752700000005E+006
10157.00         7.89704625E+006
171154.00        7.8968145E+006
79373.00         7.896186E+006
113808.00        7.8933538799999999E+006
27901.00         7.892952E+006
128820.00        7.8928827199999997E+006
25891.00         7.8905112000000002E+006
122819.00        7.8888810199999996E+006
154731.00        7.8883013300000001E+006
101674.00        7.8793245999999996E+006
51968.00         7.87910221E+006
72073.00         7.8777361100000003E+006
5182.00          7.8745217299999995E+006

1048 rows processed.
```

Query 12

```
-- TPC-H/TPC-R Shipping Modes and Order Priority Query (Q12)

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 13

```
-- TPC-H/TPC-R Customer Distribution Query (Q13)

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 14

```
-- TPC-H/TPC-R Promotion Effect Query (Q14)

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
1.6380778626395539E+001
```

1 row processed.

Query 15

```
-- TPC-H/TPC-R Top Supplier Query (Q15) Variant A

with revenue  as (
                select
                        l_suppkey supplier_no,
                        sum(l_extendedprice * (1-l_discount))
total_revenue
                from
                        lineitem
                where
                        l_shipdate >= date '1996-01-01'
                        and l_shipdate < date '1996-01-01' +
interval '3' month
                group by
                        l_suppkey
)
select
        s_suppkey,
        s_name,
        s_address,
        s_phone,
        total_revenue
from
        supplier,
        revenue
where
        s_suppkey = supplier_no
        and total_revenue = (
                select
                        max(total_revenue)
                from
                        revenue
        )
order by
        s_suppkey

S_SUPPKEY        S_NAME
S_ADDRESS                        S_PHONE        TOTAL_REVENUE
8449.00          Supplier#000008449
Wp34zim9qYFbVctdW                20-469-856-8873
1.7726272087000003E+006
```

1 row processed.

Query 16

```
-- TPC-H/TPC-R Parts/Supplier Relationship Query (Q16)

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

```
s_comment like '%Customer%Complaints%'
)
group by
p_brand,
p_type,
p_size
order by
supplier_cnt desc,
p_brand,
p_type,
p_size
```

P_BRAND	P_TYPE	P_SIZE	SUPPLIER_CNT
Brand#41	MEDIUM BRUSHED TIN	3.00	28.00
Brand#54	STANDARD BRUSHED COPPER	14.00	27.00
Brand#11	STANDARD BRUSHED TIN	23.00	24.00
Brand#11	STANDARD BURNISHED BRASS	36.00	24.00
Brand#15	MEDIUM ANODIZED NICKEL	3.00	24.00
Brand#15	SMALL ANODIZED BRASS	45.00	24.00
Brand#15	SMALL BURNISHED NICKEL	19.00	24.00
Brand#21	MEDIUM ANODIZED COPPER	3.00	24.00
Brand#22	SMALL BRUSHED NICKEL	3.00	24.00
Brand#22	SMALL BURNISHED BRASS	19.00	24.00
Brand#25	MEDIUM BURNISHED COPPER	36.00	24.00
Brand#31	PROMO POLISHED COPPER	36.00	24.00
Brand#33	LARGE POLISHED TIN	23.00	24.00
Brand#33	PROMO POLISHED STEEL	14.00	24.00
Brand#35	PROMO BRUSHED NICKEL	14.00	24.00
Brand#41	ECONOMY BRUSHED STEEL	9.00	24.00
Brand#41	ECONOMY POLISHED TIN	19.00	24.00
Brand#41	LARGE PLATED COPPER	36.00	24.00
Brand#42	ECONOMY PLATED BRASS	3.00	24.00
Brand#42	STANDARD POLISHED TIN	49.00	24.00
Brand#43	PROMO BRUSHED TIN	3.00	24.00
Brand#43	SMALL ANODIZED COPPER	36.00	24.00
Brand#44	STANDARD POLISHED NICKEL	3.00	24.00
Brand#52	ECONOMY PLATED TIN	14.00	24.00
Brand#52	STANDARD BURNISHED NICKEL	3.00	24.00
Brand#53	MEDIUM ANODIZED STEEL	14.00	24.00
Brand#14	PROMO ANODIZED NICKEL	45.00	23.00
Brand#32	ECONOMY PLATED BRASS	9.00	23.00
Brand#52	SMALL ANODIZED COPPER	3.00	23.00
Brand#11	ECONOMY BRUSHED COPPER	45.00	20.00
Brand#11	ECONOMY PLATED BRASS	23.00	20.00
Brand#11	LARGE BRUSHED COPPER	49.00	20.00
Brand#11	LARGE POLISHED COPPER	49.00	20.00
Brand#12	STANDARD ANODIZED TIN	49.00	20.00
Brand#12	STANDARD PLATED BRASS	19.00	20.00
Brand#13	ECONOMY BRUSHED BRASS	9.00	20.00
Brand#13	ECONOMY BURNISHED STEEL	14.00	20.00
Brand#13	LARGE BURNISHED NICKEL	19.00	20.00
Brand#13	MEDIUM BURNISHED COPPER	36.00	20.00
Brand#13	SMALL BRUSHED TIN	45.00	20.00
Brand#13	STANDARD ANODIZED COPPER	3.00	20.00
Brand#13	STANDARD PLATED NICKEL	23.00	20.00
Brand#14	ECONOMY ANODIZED COPPER	14.00	20.00
Brand#14	ECONOMY PLATED TIN	36.00	20.00
Brand#14	ECONOMY POLISHED NICKEL	3.00	20.00
Brand#14	MEDIUM ANODIZED NICKEL	3.00	20.00
Brand#14	SMALL POLISHED TIN	14.00	20.00
Brand#15	MEDIUM ANODIZED COPPER	9.00	20.00
Brand#15	MEDIUM PLATED TIN	23.00	20.00
Brand#15	PROMO PLATED BRASS	14.00	20.00
Brand#15	SMALL ANODIZED COPPER	45.00	20.00
Brand#15	SMALL PLATED COPPER	49.00	20.00
Brand#15	STANDARD PLATED TIN	3.00	20.00
Brand#21	LARGE ANODIZED COPPER	36.00	20.00
Brand#21	LARGE BRUSHED TIN	3.00	20.00
Brand#21	MEDIUM ANODIZED COPPER	14.00	20.00
Brand#21	PROMO BRUSHED TIN	36.00	20.00
Brand#21	PROMO POLISHED NICKEL	45.00	20.00
Brand#21	SMALL ANODIZED COPPER	9.00	20.00
Brand#21	SMALL POLISHED NICKEL	23.00	20.00
Brand#22	LARGE ANODIZED COPPER	36.00	20.00
Brand#22	LARGE BRUSHED COPPER	49.00	20.00
Brand#22	PROMO ANODIZED TIN	49.00	20.00
Brand#22	PROMO POLISHED BRASS	45.00	20.00
Brand#22	SMALL BURNISHED STEEL	45.00	20.00
Brand#23	MEDIUM ANODIZED STEEL	45.00	20.00
Brand#23	PROMO POLISHED STEEL	23.00	20.00
Brand#23	STANDARD BRUSHED TIN	14.00	20.00
Brand#23	STANDARD PLATED NICKEL	36.00	20.00
Brand#24	PROMO PLATED COPPER	49.00	20.00
Brand#24	PROMO PLATED STEEL	49.00	20.00
Brand#24	PROMO POLISHED STEEL	9.00	20.00
Brand#24	STANDARD BRUSHED TIN	36.00	20.00
Brand#25	LARGE ANODIZED BRASS	3.00	20.00
Brand#25	PROMO BURNISHED TIN	3.00	20.00
Brand#31	ECONOMY POLISHED NICKEL	3.00	20.00
Brand#31	MEDIUM PLATED TIN	45.00	20.00
Brand#31	SMALL ANODIZED STEEL	14.00	20.00
Brand#32	ECONOMY ANODIZED COPPER	36.00	20.00
Brand#32	ECONOMY BRUSHED NICKEL	49.00	20.00
Brand#32	LARGE ANODIZED TIN	19.00	20.00
Brand#32	MEDIUM BURNISHED COPPER	19.00	20.00
Brand#32	SMALL ANODIZED STEEL	45.00	20.00
Brand#33	ECONOMY POLISHED COPPER	19.00	20.00
Brand#33	PROMO PLATED NICKEL	14.00	20.00
Brand#33	SMALL POLISHED TIN	9.00	20.00
Brand#33	STANDARD ANODIZED BRASS	49.00	20.00
Brand#33	STANDARD BURNISHED BRASS	45.00	20.00
Brand#34	ECONOMY BRUSHED NICKEL	49.00	20.00
Brand#34	LARGE BRUSHED BRASS	19.00	20.00
Brand#34	SMALL BRUSHED TIN	3.00	20.00
Brand#34	STANDARD PLATED COPPER	9.00	20.00
Brand#35	LARGE ANODIZED NICKEL	3.00	20.00
Brand#35	MEDIUM ANODIZED BRASS	45.00	20.00
Brand#35	MEDIUM ANODIZED STEEL	23.00	20.00
Brand#35	PROMO ANODIZED COPPER	49.00	20.00
Brand#35	SMALL POLISHED COPPER	14.00	20.00
Brand#41	LARGE ANODIZED STEEL	3.00	20.00
Brand#41	LARGE BRUSHED NICKEL	23.00	20.00
Brand#41	LARGE BURNISHED COPPER	3.00	20.00
Brand#41	MEDIUM PLATED STEEL	19.00	20.00
Brand#41	SMALL BURNISHED COPPER	23.00	20.00
Brand#42	MEDIUM BURNISHED BRASS	14.00	20.00
Brand#42	SMALL BURNISHED COPPER	3.00	20.00
Brand#43	ECONOMY POLISHED COPPER	9.00	20.00
Brand#43	SMALL PLATED STEEL	3.00	20.00
Brand#43	STANDARD BURNISHED TIN	23.00	20.00
Brand#44	LARGE ANODIZED STEEL	23.00	20.00
Brand#44	PROMO ANODIZED TIN	23.00	20.00
Brand#51	ECONOMY BRUSHED BRASS	49.00	20.00
Brand#51	ECONOMY POLISHED NICKEL	9.00	20.00
Brand#51	MEDIUM BRUSHED TIN	9.00	20.00
Brand#51	MEDIUM PLATED BRASS	9.00	20.00
Brand#51	PROMO BURNISHED BRASS	9.00	20.00
Brand#51	SMALL PLATED NICKEL	49.00	20.00
Brand#51	STANDARD ANODIZED NICKEL	49.00	20.00
Brand#51	STANDARD BRUSHED COPPER	3.00	20.00
Brand#52	ECONOMY ANODIZED BRASS	3.00	20.00
Brand#52	ECONOMY BRUSHED COPPER	49.00	20.00
Brand#52	LARGE ANODIZED NICKEL	45.00	20.00

------ rows deleted ------

Brand#55	STANDARD BRUSHED COPPER	14.00	4.00
Brand#55	STANDARD BRUSHED COPPER	19.00	4.00
Brand#55	STANDARD BRUSHED NICKEL	3.00	4.00
Brand#55	STANDARD BRUSHED NICKEL	36.00	4.00
Brand#55	STANDARD BRUSHED STEEL	9.00	4.00
Brand#55	STANDARD BRUSHED STEEL	14.00	4.00
Brand#55	STANDARD BRUSHED STEEL	19.00	4.00
Brand#55	STANDARD BRUSHED STEEL	49.00	4.00
Brand#55	STANDARD BRUSHED TIN	19.00	4.00
Brand#55	STANDARD BRUSHED TIN	49.00	4.00
Brand#55	STANDARD BURNISHED BRASS	9.00	4.00
Brand#55	STANDARD BURNISHED BRASS	19.00	4.00

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

18314 rows processed.

Query 17

```
-- TPC-H/TPC-R Small-Quantity-Order Revenue Query (Q17)

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
3.4840605428571434E+005


1 row processed.
```

Query 18

```
-- TPC-H/TPC-R Large Volume Customer Query (Q18)

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#0000128120       128120.00           4722021.00
1994-04-07
5.4408908999999997E+005 3.23E+002
Customer#0000144617       144617.00           3043270.00
1997-02-12
5.3060443999999994E+005 3.17E+002
Customer#0000013940       13940.00            2232932.00
1997-04-13
5.2272060999999999E+005 3.04E+002
Customer#0000066790       66790.00            2199712.00
1996-09-30
5.1553182000000001E+005 3.27E+002
```

Customer#0000046435 46435.00 4745607.00
1997-07-03
5.0804798999999999E+005 3.09E+002
Customer#0000015272 15272.00 3883783.00
1993-07-28
5.0024133000000002E+005 3.02E+002
Customer#0000146608 146608.00 3342468.00
1994-06-12
4.9979458000000002E+005 3.03E+002
Customer#0000096103 96103.00 5984582.00
1992-03-16
4.9439878999999998E+005 3.12E+002
Customer#0000024341 24341.00 1474818.00
1992-11-15
4.9134826000000001E+005 3.02E+002
Customer#0000137446 137446.00 5489475.00
1997-05-23
4.8776325E+005 3.11E+002
Customer#0000107590 107590.00 4267751.00
1994-11-04
4.8514138E+005 3.01E+002
Customer#0000050008 50008.00 2366755.00
1996-12-09
4.8389126000000001E+005 3.02E+002
Customer#0000015619 15619.00 3767271.00
1996-08-07
4.8008396000000002E+005 3.18E+002
Customer#0000077260 77260.00 1436544.00
1992-09-12
4.7949942999999999E+005 3.07E+002
Customer#0000109379 109379.00 5746311.00
1996-10-10
4.7806410999999999E+005 3.02E+002
Customer#0000054602 54602.00 5832321.00
1997-02-09
4.7122008000000002E+005 3.07E+002
Customer#0000105995 105995.00 2096705.00
1994-07-03
4.6969258000000002E+005 3.07E+002
Customer#0000148885 148885.00 2942469.00
1992-05-31
4.6963044E+005 3.13E+002
Customer#0000114586 114586.00 551136.00
1993-05-19
4.6960559000000003E+005 3.08E+002
Customer#0000105260 105260.00 5296167.00
1996-09-06
4.6936057000000001E+005 3.03E+002
Customer#0000147197 147197.00 1263015.00
1997-02-02
4.6714966999999998E+005 3.2E+002
Customer#0000064483 64483.00 2745894.00
1996-07-04
4.6699134999999998E+005 3.04E+002
Customer#0000136573 136573.00 2761378.00
1996-05-31
4.6128272999999998E+005 3.01E+002
Customer#0000016384 16384.00 502886.00
1994-04-12
4.5837891999999998E+005 3.12E+002
Customer#0000117919 117919.00 2869152.00
1996-06-20
4.5681591999999998E+005 3.17E+002
Customer#0000012251 12251.00 735366.00
1993-11-24
4.5510726000000001E+005 3.09E+002
Customer#0000120098 120098.00 1971680.00
1995-06-14
4.5345122999999998E+005 3.08E+002
Customer#0000066098 66098.00 5007490.00
1992-08-07
4.5343615999999997E+005 3.04E+002
Customer#0000117076 117076.00 4290656.00
1997-02-05
4.4954584999999998E+005 3.01E+002
Customer#0000129379 129379.00 4720454.00
1997-06-07
4.4866578999999998E+005 3.03E+002
Customer#0000126865 126865.00 4702759.00
1994-11-07
4.4760665000000002E+005 3.2E+002
Customer#0000088876 88876.00 983201.00
1993-12-30
4.4671746000000002E+005 3.04E+002
Customer#0000036619 36619.00 4806726.00
1995-01-17
4.4670409000000003E+005 3.28E+002
Customer#0000141823 141823.00 2806245.00
1996-12-29
4.4626912E+005 3.1E+002
Customer#0000053029 53029.00 2662214.00
1993-08-13
4.4614448999999999E+005 3.02E+002
Customer#0000018188 18188.00 3037414.00
1995-01-25
4.4380721999999997E+005 3.08E+002
Customer#0000066533 66533.00 29158.00
1995-10-21
4.435765E+005 3.05E+002
Customer#0000037729 37729.00 4134341.00
1995-06-29
4.4108296999999997E+005 3.09E+002
Customer#0000003566 3566.00 2329187.00
1998-01-04
4.3980335999999999E+005 3.04E+002
Customer#0000045538 45538.00 4527553.00
1994-05-22
4.3627531E+005 3.05E+002
Customer#0000081581 81581.00 4739650.00
1995-11-04
4.3540590000000002E+005 3.05E+002
Customer#0000119989 119989.00 1544643.00
1997-09-20
4.3456825E+005 3.2E+002
Customer#0000003680 3680.00 3861123.00
1998-07-03
4.3352596999999997E+005 3.01E+002
Customer#0000113131 113131.00 967334.00
1995-12-15
4.3295775E+005 3.01E+002
Customer#0000141098 141098.00 565574.00
1995-09-24
4.3098669E+005 3.01E+002
Customer#0000093392 93392.00 5200102.00
1997-01-22
4.2548751000000001E+005 3.04E+002
Customer#0000015631 15631.00 1845057.00
1994-05-12
4.1987959000000003E+005 3.02E+002
Customer#0000112987 112987.00 4439686.00
1996-09-17
4.1816148999999999E+005 3.05E+002
Customer#0000012599 12599.00 4259524.00
1998-02-12
4.1520060999999999E+005 3.04E+002
Customer#0000105410 105410.00 4478371.00
1996-03-05
4.1275451000000001E+005 3.02E+002
Customer#0000149842 149842.00 5156581.00
1994-05-30
4.1132934999999998E+005 3.02E+002
Customer#0000010129 10129.00 5849444.00
1994-03-21
4.0912984999999998E+005 3.09E+002
Customer#0000069904 69904.00 1742403.00
1996-10-19
4.08513E+005 3.05E+002

```
Customer#0000017746     17746.00        6882.00         1997-
04-09
4.0844692999999999E+005 3.03E+002
Customer#0000013072     13072.00        1481925.00
1998-03-15
3.9919546999999997E+005 3.01E+002
Customer#0000082441     82441.00        857959.00
1994-02-07
3.8257973999999999E+005 3.05E+002
Customer#0000088703     88703.00        2995076.00
1994-01-30
3.6381212E+005 3.02E+002


57 rows processed.
```

Query 19

```
-- TPC-H/TPC-R Discounted Revenue Query (Q19)

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
3.0838430578000001E+006


1 row processed.
```

Query 20

```
-- TPC-H/TPC-R Potential Part Promotion Query (Q20)

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

S_NAME                  S_ADDRESS
Supplier#000000020      iybAE,RmTymrZVYaFZva2SH,j
Supplier#000000091
YV45D7TkfdQanOOZ7q9QxkyGUapU1oOWU6q3
Supplier#000000197      YC2Acon6kjY3zj3Fbxs2k4Vdf7X0cd2F
Supplier#000000226      83qOdU2EYRdPQAQhEtn GRZEd
Supplier#000000285      Br7e1nnt1yxrw6ImgpJ7YdhFDjuBf
Supplier#000000378      FfbhyCxWvcPrO8ltp9
Supplier#000000402      i9Sw4DoyMhzhKXCH9By,AYSgmD
Supplier#000000530      0qwCMwobKY OcmLyfRXlagA8ukENJv,
Supplier#000000688      D fw5ocppmZpYBBIPI718hCihLDZ5KhKX
Supplier#000000710      f19YPvOyb QoYwjKC,oPycpGfieBAcwKJo
Supplier#000000736
l6i2nMwVuovfKnuVgaSGK2rDy65DlAFLegiL7
Supplier#000000761      zlSLelQUj2XrvTTFnv7WAcYZGvvMTx882d4
Supplier#000000884      bmhEShejaS
Supplier#000000887      urEaTejH5POADP2ARrf
Supplier#000000935      ij98czM 2KzWe7dDTOxB8sq0UfCdvrX
Supplier#000000975      ,AC e,tBpNwKb5xMUzeohxlRn,
hdZJo73gFQF8y
Supplier#000001263      rQWr6nf8ZhB2TAiIDIvo5Io
Supplier#000001399      LmrocnIMSyYOWuANx7
Supplier#000001446      lch9HMNU1R7a0LIybsUodVknk6
Supplier#000001454      TOpimgu2TVXIjhiL93h,
Supplier#000001500      wDmF5xLxtQch9ctVu,
Supplier#000001602      uKNWIeafaM644
Supplier#000001626      UhxNRzUu1dtFmp0
Supplier#000001682      pXTkGxrTQVyH1Rr
Supplier#000001699      Q9C4rfJ26oijVPqqcqVXeRI
Supplier#000001700      7hMICof1Y5zLFg
Supplier#000001726      TeRY7TtTH24sEword7yAaSkjx8
Supplier#000001730      Rc8e,1Pybn r6zo0VJIEiD0UD vhk
Supplier#000001746
qWsendlOekQG1aW4uq06uQaCm51se8lirv7 hBRd
Supplier#000001752      Fra7outx41THYJaRThdOGiBk
Supplier#000001856
jXcRgzYF0ah05iR8p6w5SbJJLcUGyYiURPvFwUWM
Supplier#000001931      FpJbMU2h6ZR2eBv8I9NIxF
Supplier#000001939      Nrk,JA4bfReUs
Supplier#000001990
DSDJkCgBJzuPg1yuM,CUdLnsRliOxkkHezTCA
```

```
------ rows deleted ------

POeheRMdj9tmpyeQ,BfCXN5BIAb
Supplier#000008366
h778cEj14BuW9OEKlvPTWq4iwASR6EBBXN7zeS8
Supplier#000008423      RQhKnkAhR0DAr3Ix4Q1weMMn00hNe Kq
Supplier#000008480      4sSDA4ACReklNjEm5T6b
Supplier#000008532      Uc29q4,5xVdDOF87UZrxhr4xWS0ihEUXuh
Supplier#000008595      MH0iB73GQ3z UW3O DbCbqmc
Supplier#000008610      SgVgP90vP452sUNTgzL9zKwXHXAzV6tV
Supplier#000008705      aE,trRNdPx,4yinTD9O3DebDIp
Supplier#000008742      HmPlQEzKCPEcTUL14,kKq
Supplier#000008841      I 85Lu1sekbg2xrSIzm0
Supplier#000008895      2cH4okfaLSZTTg8sKRbbJQxkmeFu2Esj
Supplier#000008967      2kwEHyMG
7FwozNImAUE6mH0hYtqYculJM
Supplier#000008972      w2vF6 D5YZO3visPXsqVfLADTK
Supplier#000009032      qK,trB6Sdy4Dz1BRUFNy
Supplier#000009147      rOAuryHxpZ9eOvx
Supplier#000009252      F7cZaPUHwh1 ZKyj3xmAVWC1XdP
ue1p5m,i
Supplier#000009278      RqYTzgxj93CLX 0mcYfCENOefD
Supplier#000009327      uoqMdf7e7Gj9dbQ53
Supplier#000009430      igRqmneFt
Supplier#000009567      r4Wfx4c3xsEAjcGj71HHZByornl D9vrztXlv4
Supplier#000009601      51m637bO,Rw5DnHWFUvLacRx9
Supplier#000009709      rRnCbHYgDgl9PZYnyWKVYSUW0vKg
Supplier#000009753      wLhVEcRmd7PkJF4FBnGK7Z
Supplier#000009796      z,y4Idmr15DOvPUqYG
Supplier#000009799       4wNjXGa4OKWl
Supplier#000009811      E3iuyq7UnZxU7oPZIe2Gu6
Supplier#000009812      APFRMy3lCbgFga53n5t9DxzFPQPgnjrGt32
Supplier#000009862      rJzweWeN58
Supplier#000009868      ROjGgx5gvtkmnUUoeyy7v
Supplier#000009869
ucLqxzrpBTRMewGSM29t0rNTM30g1Tu3Xgg3mKag
Supplier#000009899      7XdpAHrzr1t,UQFZE
Supplier#000009974      7wJ,J5DKcxSU4Kp1cQLpbcAvB5AsvKT


204 rows processed.
```

Query 21

```
-- TPC-H/TPC-R Suppliers Who Kept Orders Waiting Query (Q21)

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

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

------ rows deleted ------

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 22

```
-- TPC-D Global Sales Opportunity Query (Q22)

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             6.7377139900000049E+006
17        861.00             6.4605737200000035E+006
18        964.00             7.2366873999999994E+006
23        892.00             6.701457950000003E+006
29        948.00             7.158866629999999E+006
30        909.00             6.8084361299999999E+006
31        922.00             6.8066701799999969E+006
```

7 rows processed.

Appendix - D
Benchmark Scripts

D.1 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;
```

D.2 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;
```

D.3 gen_seed.sh

```
#!/bin/ksh

SEED_FILE=$1

#Generate the seed
echo "Setting the random number seed"
PSEED=`date +%m:%d:%H:%M:%S | sed -e 's/://g'`
echo "Using ${PSEED} as seed0"
echo ${PSEED} >  $SEED_FILE
echo "Done setting the random number seed"
```

D.4 gtime.c

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

# include <sys/time.h>

main ()
{

struct timeval tp ;

  gettimeofday(&tp, NULL) ;

  printf ("%ld\n", tp.tv_sec) ;
/*
  printf ("usec = %ld\n", tp.tv_usec) ;
*/
}

/*
double gettime();
main() {
  printf("%f", gettime());
  exit(0);
}
*/
```

D.5 qexecpl.c

```
#ifdef RCSID

static char *RCSid =
   "$Header: qexecpl.c 22-feb-2001.11:48:08 mpoess Exp $ ";
#endif /* RCSID */

/* Copyright (c) Oracle Corporation 1999. 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    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 "qexecpl.h"

/* Function Prototypes */

extern double gettime();

/* function prototypes from gen.c */

int get_statement();

/* Declare error handling functions */

void sql_error();

/* Other prototypes */

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

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

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

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

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

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

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

double product = 1.0;     /* cumulative product of query times   */
int rows_ret = 0;         /* the number of rows fetched          */
```

```
int num_sel_list = 0;     /* the number of select list item      */

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

sltype slist[MAX_SEL_LIST];  /* Array for describing Select List        */
dltype *dlist[MAX_SEL_LIST]; /* Array of ptrs for Defining Select List
*/

char stmt[SQL_LEN];       /* The SQL statement or comment line.  */
char cmnt[81];            /* Buffer to save the comment.         */
#ifdef LINUX
FILE *qtemp;      /* fd for query template               */
FILE *logfile;   /* log and report files                */
FILE *rep;
#else
FILE *qtemp = stdin;      /* fd for query template               */
FILE *logfile = stdout;   /* log and report files                */
FILE *rep = stdout;
#endif
void *defbuf;             /* Buffer pointer for ODEFIN           */
int deflen = 0;           /* Size of data type for ODEFIN        */
int deftype = 1;          /* Oracle type number for ODEFIN       */

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

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

/* OCI handles */

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

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

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

void usage() {

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



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

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

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

  /* Rollback just in case */

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

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

  SQLexit();

  exit(1);
}

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

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

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

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

  strcpy(logname, argv[1]);

  /* Process  optional parameters */
```

```
argc -= 1;
argv += 1;

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

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

SQLinit();

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

time(&tim);
fprintf(logfile, "Begin Execution at %s\n\n", ctime(&tim));

/* Get the next statement and start processing it */

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

  switch (retcode) {

    /* If this is a comment, skips it */
  case COMMENT:
    if (end_flag) {
            end_flag = 0;      /* reset query end flag */
            /* save the comment so that we can print it out later on  */
            strcpy(cmnt, stmt);
            break;
    }
    fprintf(logfile, "%s", stmt);
    fprintf(rep, "%s", stmt);
    break;

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

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

    /* Executes the query */

    SQLexec();

      s_tr_end = gettime();
      stmt_cnt++;

      /*
      fprintf(logfile,"\nStatement Started at %.2f\n", s_tr_start);
      fprintf(logfile,"Statement Ended at %.2f\n", s_tr_end);
      */
      fprintf(logfile,"Statement Processed in %.2f seconds.\n",
                  (s_tr_end - s_tr_start));
      break;

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

  /* Get Timing for the last query */

  tr_end = gettime();

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

  fprintf(rep, "%.2f\n", (tr_end - tr_start));

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

  fflush(rep);
  fflush(logfile);

  /* Close the query template file */

  fclose(qtemp);

  /*  Disconnect from ORACLE. */

  SQLexit();
  exit(0);
}



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

void SQLinit() {

  int i;

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

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

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

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

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

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

  /* get username and password */

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

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


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

OCIaset(tpcusr,OCI_HTYPE_SESSION,logname,strlen(logname),OCI
_ATTR_USERNAME,
                errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI_A
TTR_PASSWORD,
                errhp);

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


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

  /* Enable session parallel dml */

  sprintf((char *) stmt, PDMLTXT);

  OCIStmtPrepare(cur_dml,errhp,(text *)stmt,strlen((char *)stmt),
              OCI_NTV_SYNTAX,OCI_DEFAULT);
  OCIsexec(tpcsvc,cur_dml,errhp,1);

  /* Enable session parallel ddl */

  sprintf((char *) stmt, PDDLTXT);

  OCIStmtPrepare(cur_ddl,errhp,(text *)stmt,strlen((char *)stmt),
              OCI_NTV_SYNTAX,OCI_DEFAULT);
  OCIsexec(tpcsvc,cur_ddl,errhp,1);

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



/* SQLexec()  Executes the SQL statement.                          */
/*            Parse the SQL statement.                             */
/*            If DDL or DML statements, execute right away.        */
/*            Else describe and define select list outputs,        */
/*                 execute and fetch results.                      */

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

  if (!end_flag) {

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

    tr_end = gettime();

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

      fprintf(logfile,"-------------------------------------------------\n\n");

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

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

      fprintf(rep, "%.2f\n", (tr_end - tr_start));
      fprintf(rep, "%s", cmnt);

      fprintf(logfile,"\nBegan Executing this Query at %s\n", ctime(&tim));

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

      fflush(logfile);
      fflush(rep);

    }

    tr_start = tr_end;
    qry_cnt++;

    end_flag = 1;
  }

  s_tr_start = gettime();

  /* prepare the statement */

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

  /* Prints the query text to the logfile */

  fprintf(logfile, "\n%s\n", stmt);

  /* if this is a DDL or DML statement, execute it right away */
  /* only worries about SELECT statements right now, cannot   */
```

```
  /* execute a stored PL/SQL procedure in thie version        */


OCIaget(curq,OCI_HTYPE_STMT,&stmttyp,NULL,OCI_ATTR_STMT_
TYPE,errhp);

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

  }

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

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

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

  OCIsexec(tpcsvc,curq,errhp,0);

  num_sel_list = define_output_variables();

  /* Executes the query and fetches the rows */

  (void) process_select_list(num_sel_list);

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


OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,OCI_ATTR_ROW
_COUNT,errhp);

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

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

  /* reset set_fetchrows */

  num_to_fetch = -1;

}


void SQLexit() {

  int i;

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

  /* free all memory */

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

  /* Flush all output */

  fflush(rep);
  fflush(logfile);

}


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

int define_output_variables()
{

  int i;
  int retflag = 0;

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

    slist[i].buflen = MAX_COLNAME_SIZE;

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


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

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

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

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

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

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

    switch (slist[i].dbtype) {

    case OCI_TYPECODE_NUMBER:

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

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

```
      deflen = FLT;
      deftype = OCI_TYPECODE_FLOAT;
      slist[i].dbtype = OCI_TYPECODE_FLOAT;
#endif /* HAVE_SCALE */

      break;

    default:

      /* default is character string */

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

    /* Define the column */

    if ((status=OCIDefineByPos(curq,&(dlist[i]->defhdl),errhp,POS(i),
                                        defbuf,deflen,deftype,NULL,
                                        dlist[i]-
>rlen,NULL,OCI_DEFAULT))!=OCI_SUCCESS)
      sql_error(errhp,status,1);
  }
  return i;
}


/* process_select_list(): Fetch rows from a query.                */

void process_select_list(num)
    int num;        /* number of select list items */
{

  int i,j;
  int ntf;
  int num_so_far;
  sword stats = OCI_SUCCESS;

  /* Print the headers for the query execution result */

  print_header(num);

  /* See if we need to limit the rows to fetch */

  ntf = (num_to_fetch >= 0) ? num_to_fetch : MAX_ARRAY;

  /* Fetch the rows and print them out */

  if ((ntf > MAX_ARRAY) || (num_to_fetch == -1)) {

    stats = OCIStmtFetch(curq, errhp, MAX_ARRAY,
OCI_FETCH_NEXT, OCI_DEFAULT);


OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,OCI_ATTR_ROW
_COUNT,errhp);

    print_rows(num,rows_ret);

    /* To avoid 1022 from OFEN */
    /* More rows to fetch...   */

    if (stats != OCI_NO_DATA) {
      if (num_to_fetch == -1) {
            while ((stats =
OCIStmtFetch(curq,errhp,MAX_ARRAY,OCI_FETCH_NEXT,

                                                OCI_DEFAULT)) ==
OCI_SUCCESS) {
              OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                          OCI_ATTR_ROW_COUNT,errhp);
              print_rows(num,(num_so_far-rows_ret));
              rows_ret = num_so_far;
            }
            /* Print the final rows */
            OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                        OCI_ATTR_ROW_COUNT,errhp);
            print_rows(num,(num_so_far-rows_ret));
            rows_ret = num_so_far;
      } else {
            ntf -= MAX_ARRAY;

            while ((stats = OCIStmtFetch(curq,errhp,
                                                ((ntf>MAX_ARRAY) ?
MAX_ARRAY:ntf),
                                                OCI_FETCH_NEXT,
OCI_DEFAULT)) ==
                    OCI_SUCCESS) {
              ntf -= MAX_ARRAY;
              OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                          OCI_ATTR_ROW_COUNT,errhp);
              print_rows(num,(num_so_far-rows_ret));
              rows_ret = num_so_far;
              if (ntf <= 0) break;
            }
            OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                        OCI_ATTR_ROW_COUNT,errhp);
            print_rows(num,(num_so_far-rows_ret));
            rows_ret = num_so_far;
      }
    }
  } else {
    OCIStmtFetch(curq, errhp, ntf, OCI_FETCH_NEXT,
OCI_DEFAULT);

OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,OCI_ATTR_ROW
_COUNT,errhp);
    print_rows(num,rows_ret);
  }

  fprintf(logfile,"\n\n%d row%c processed.\n", rows_ret,
              rows_ret == 1 ? '\0' : 's');

}


int get_statement()
{

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

  /* Reset statement buffer */

  stmt[0] = '\0';

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

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

    /* remove blanks */

    str = line;

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

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

    strcat(stmt, str);

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

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

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


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

void *memalloc(size)
     int size;
{

  void *tmp;

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



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

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

  fprintf(logfile, "\n");

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

    /* extract the column name */

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

    /* format the output a little */

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

    /* do a little bit of formatting */

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

  fprintf(logfile,"\n");
}


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

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

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

    len = 0;

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

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


      /* do a little bit of formatting */

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

      switch(slist[j].dbtype) {
      case INT_TYPE:

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


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

void remove_newline(str)
     char *str;
{

  char *p;

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

D.6 qexecpl.h

```
/*

 * $Header: qexecpl.h 22-feb-2001.11:50:33 mpoess Exp $
 */

/* Copyright (c) Oracle Corporation 1999. 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      02/22/01 - add linux changes
   mpoess      08/05/99 - make compile
   mpoess      07/15/99 - Creation
   mpoess      07/15/99 - Creation

*/
#define grep HAVE_SCALE 1
/*
# ifndef S_ORACLE
#  include <s.h>
# endif
*/
#ifndef QSTREAMPL_H

#define QSTREAMPL_H

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

#include <oratypes.h>

#include <oratypes.h>

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

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

/* some basic definitions */

#define UNAME_LEN 64
#define MAX_FILE_PATH_LEN 128

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

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

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

/* defines and typedefs for query description */

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

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

/* types for describe */

#define CHAR_TYPE 1
#define NUM_TYPE  2
#define INT_TYPE  3
#define FLT_TYPE  4
#define STR_TYPE  5
#define DATE_TYPE 12

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

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

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


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

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

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

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

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

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

extern int errno;

#define SQL_LEN 2048

#ifndef NULL
#define NULL 0
#endif

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

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#ifndef ub1
#define ub1 unsigned char
#endif

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

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

/* For get_statement */

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


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

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

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

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

#define OCIsexec(svch,stmh,errh,iter) \

if((status=OCIStmtExecute(svch,stmh,errh,iter,0,NULL,NULL,OCI_DEF
AULT)) != 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
160)"
#define PDDLTXT "alter session force parallel ddl parallel (degree 64)"

#endif /* QSTREAMPL_H */
```

D.7 runTPCHall

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

ECHO=echo

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

RUN_ID_FILE=${KIT_DIR}/audit/r_id

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

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

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

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

echo Start TPC-H Benchmark SEQUENCE NUMBER: $RUN_ID >
$SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE
```

```
echo "Starting a new Oracle log file:
$ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log" >>
$SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE

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

echo "Start: load database `date`" >> $SCRIPT_LOG_FILE
dbcre_3k.sh > $LD1DBCRE
sctso_3k.sh > $LD2SCTSO
STIME=`$GTIME`
echo "Start: timed load portion `date`" >> $SCRIPT_LOG_FILE
dapop_3k.sh > $LD3DAPOP
ixcre_3k.sh > $LD4IXCRE
anlyz_3k.sh > $LD5ANLYZ
echo "End: timed load portion `date`" >> $SCRIPT_LOG_FILE

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

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

ckpnt.sh
runTPCHpt ${SCALE_FACTOR} 1 ${RUN_ID}
ckpnt.sh
runTPCHpt ${SCALE_FACTOR} 2 ${RUN_ID}

sleep 600
tshut >> $SCRIPT_LOG_FILE

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

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

D.8 mount_flats.sh

```
#!/bin/ksh
runas root mount –t flats
```

D.9 umount_flats.sh

```
#!/bin/ksh
runas root umount –t flats
```

D.10 runTPCHpt

```
#!/bin/ksh
. $KIT_DIR/env
#set -x
#ECHO=/bin/echo
SCRIPT_DIR=${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=${KIT_DIR}/dbgen
#/home/oracle/tpch/kit/dbgen
QGEN=${QGEN_DIR}/qgen
QEXEC=${SRC_DIR}
DSS_QUERY=${KIT_DIR}/queries
export DSS_QUERY

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

TPCD_BIN=${KIT_DIR}/audit/bin

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

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

# The defaults

QPROG=${QEXEC}/qexec

usage () {

echo " "
echo "Usage: $0 [-p <program for query stream>] [-u1 <program for
UF1>]"
echo "          [-u2 <program for UF2>] [-o] [-s] [-h] [-u <user/password>]"
echo "          <scale factor> <run_number>"
echo ""
echo "scale factor     : The scale factor of the run."
echo "update ||ism     : The parallelism to use for the UFs."
echo ""
echo "-p <program>     : Program for Query Stream."
echo "                   Default is $QPROG."
echo "-u1 <program>    : Program for UF1."
echo "                   Default is $U1PROG."
echo "-u2 <program>    : Program for UF2."
echo "                   Default is $U2PROG."
echo "-o               : Collect Oracle statistics."
echo "-s               : Collect System statistics."
echo "-u <user/passwd> : User/Password. Default is tpch/tpch."
echo "-h               : Displays this message."
}
set -- `getopt "p:1:2:osu:h" "$@"` || usage

while :
do
  case "$1" in
  -1) shift; U1PROG=$1;;
  -2) shift; U2PROG=$1;;
  -p) shift; QPROG=$1;;
  -o) OSTAT=1;;
  -s) SSTAT=1;;
  -h) usage; exit 0;;
  --) shift; break;;
  esac
  shift;
done


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

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

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

```
        mkdir $OUT_DIR
fi

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

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

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


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

echo >> $SCRIPT_LOG_FILE

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

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

# Execute UF1

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

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

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

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

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

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

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

# Execute UF2

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

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

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

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


echo "End UF2 $UF2_END, $EDATE" >>  $SCRIPT_LOG_FILE
echo UF2: Execution Time: $UF2_TIME >> $SCRIPT_LOG_FILE
echo >>  $SCRIPT_LOG_FILE

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

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

i=$START_SET
PSEED=`cat $SEED_FILE`

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

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

  i=`expr $i + 1`
done

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

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

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

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

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

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

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

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

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

D.11 runTPCHus

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

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

DSS_QUERY=${KIT_DIR}/queries
export DSS_QUERY

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

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


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

GTIME=${SRC_DIR}/gtime
HID=1

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


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

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

  # Execute UF1

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

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

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

  # Execute UF2

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

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

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

print > /tmp/th_pipe2
```

D.12 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.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      10/25/01 - change default directory for update sets
#    mpoess      10/17/01 - add support for external tables
```

```
#    mpoess     08/15/99 - Creation
#    mpoess     08/15/99 - Creation
#
#
. $KIT_DIR/env
O=${ORACLE_HOME}
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
UTILS_DIR=${KIT_DIR}/utils
LOG_DIR=${UPDATE_DIR}/log
GTIME=${UTILS_DIR}/gtime
SF=${SCALE_FACTOR}
#PAR_HINT=${UPDATE_DOP}
PAR_HINT_ET=4
PAR_HINT=128

LOGPATH=.
PASSWD=${DATABASE_USER}

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

# perform the update function 1

START=`$GTIME`

# first create the temp tables

sqlplus /NOLOG << !

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

drop directory rf_dir;
create directory rf_dir as '/rfdata/3000';

drop table temp_l_et;
create table temp_l_et(
    l_orderkey          number ,
    l_partkey           number ,
    l_suppkey           number ,
    l_linenumber        number ,
    l_quantity          binary_double ,
    l_extendedprice     binary_double ,
    l_discount          binary_double ,
    l_tax               binary_double ,
    l_returnflag        char(1) ,
    l_linestatus        char(1) ,
    l_shipdate          date ,
    l_commitdate        date ,
    l_receiptdate       date ,
    l_shipinstruct      char(25) ,
    l_shipmode          char(10) ,
    l_comment           varchar(44)
)
organization external (
type ORACLE_LOADER
default directory rf_dir
access parameters
(
            records delimited by newline
                        date_cache 3000
            nobadfile
            nologfile
            nodiscardfile
            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_orderkey          number ,
    o_custkey           number ,
    o_orderstatus       char(1) ,
    o_totalprice        binary_double ,
    o_orderdate         date ,
    o_orderpriority     char(15) ,
    o_clerk             char(15) ,
    o_shippriority      number ,
    o_comment           varchar(79)
)
organization external (
type ORACLE_LOADER
default directory rf_dir
access parameters
(
            records delimited by newline
                        date_cache 3000
            nobadfile
            nologfile
            nodiscardfile
            fields terminated by '|'
            missing field values are null
        )
        location (
        'orders.tbl.u${SETNUM}'
))
reject limit unlimited;
alter table temp_l_et parallel ${PAR_HINT_ET};
alter table temp_o_et parallel ${PAR_HINT_ET};

alter session force parallel dml parallel (degree ${PAR_HINT});
alter session set isolation_level = serializable;
alter session set optimizer_index_cost_adj = 1;
-- alter session set events '10979 trace name context forever, level 1';
-- alter session set events '10132 trace name context forever';
commit;

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

insert into lineitem (
  select
    l_shipdate          ,
    l_orderkey          ,
    l_discount          ,
    l_extendedprice     ,
    l_suppkey           ,
    l_quantity          ,
    l_returnflag        ,
    l_partkey           ,
    l_linestatus        ,
    l_tax               ,
    l_commitdate        ,
    l_receiptdate       ,
    l_shipmode          ,
```

```
    l_linenumber        ,
    l_shipinstruct      ,
    l_comment
from temp_l_et);
commit;

drop table temp_l_et;
drop table temp_o_et;

exit;
!

END=`$GTIME`

# Done

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

echo ""
```

D.13 runuf2.sh

```
#!/bin/ksh
#
# $Header: runuf2.sh 25-oct-2001.15:56:05 mpoess Exp $
#
# runuf2.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation.  All rights reserved.
#
#    NAME
#      runuf2.sh - <one-line expansion of the name>
#
# DESCRIPTION
#    runuf2.sh [-u <uid/passwd to login>] [-p <program>] <run_id>
#              <scale factor> <pair number> <parallelism>
# USAGE
#    To execute UF2.
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      10/25/01 - change default directory for update sets
#    mpoess      10/17/01 - add support for external tables
#    mpoess      08/15/99 - Creation
#    mpoess      08/15/99 - Creation
#
. $KIT_DIR/env
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
UTILS_DIR=${KIT_DIR}/utils
GTIME=${UTILS_DIR}/gtime
LOG_DIR=${UPDATE_DIR}/log
#PAR_HINT=${UPDATE_DOP}
PAR_HINT=256
SF=${SCALE_FACTOR}
PASSWD=${DATABASE_USER}

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

SETNUM=$1

i=1
PID=""

START=`$GTIME`

# first create the temp tables

sqlplus /NOLOG << !

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

drop directory rf_dir;
create directory rf_dir as '/rfdata/3000';

drop table temp_okey_et;
drop table temp_okey;

create table temp_okey_et(
    t_orderkey          number
)
organization external (
type ORACLE_LOADER
default directory rf_dir
access parameters
(
                records delimited by newline
                nobadfile
                nologfile
                nodiscardfile
                fields terminated by '|'
                missing field values are null
        )
        location (
        'delete.${SETNUM}'))
reject limit unlimited;

alter table temp_okey_et parallel 8;
create table temp_okey parallel 32  nologging as select * from
temp_okey_et;
create unique index i_temp_okey on temp_okey (t_orderkey) parallel
32 nologging compute statistics;
analyze table temp_okey estimate statistics sample 2 percent;

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

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

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

drop table temp_okey;
drop table temp_okey_et;
exit;
!

END=`$GTIME`

# Done

echo ""
echo "Update Function 2 Set $SETNUM done!"
echo "Elapsed Time is `echo $END - $START | bc`"
echo ""
```

D.14 scnt.sh

```
#!/bin/ksh

echo Process count for TPC-H RUN:$1 SEQUENCE:$2
while [ 1 = 1 ]; do
  cnt=`ps -ef | egrep "qexec|runTPCHus" | grep -v grep | wc -l`
                echo
        echo `date` : $cnt
```

```
        ps -ef | egrep "qexec|runTPCHus" | grep -v grep
        sleep 30
done
```

D.15 stopasm

```
#!/bin/ksh

export ORACLE_SID=+ASM1
sqlplus /NOLOG <<EOF
connect /as sysdba
shutdown normal
shutdown abort
EOF
```

D.16 startasm

```
#!/bin/ksh

INITFILE=$SCHEMA_DIR/init+ASM1.ora
export ORACLE_SID=+ASM1
sqlplus /NOLOG <<EOF
connect /as sysdba
startup pfile= $INITFILE mount
EOF
```

Appendix - E ACID Scripts

E.1 a_query.sql

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

set serverout on;

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

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

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

exit;
```

E.2 a_query2.sql

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

set serverout on;

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

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

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

exit;
```

E.3 atom.sh

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

. $KIT_DIR/env

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

usage() {

   echo ""
```

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

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

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

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

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

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

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

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

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

E.4 atrans.sql

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


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

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

CREATE OR REPLACE PACKAGE BODY d_atrans
IS
PROCEDURE doatrans
(
       l_key          IN OUT integer,
       o_key          IN OUT integer,
       delta          IN OUT integer,
       l_pkey         IN OUT integer,
       l_skey         IN OUT integer,
       l_quan         IN OUT integer,
       l_newquan      IN OUT integer,
       l_tax          IN OUT number,
       l_disc         IN OUT number,
       l_eprice       IN OUT number,
       l_neweprice    IN OUT number,
       o_tprice       IN OUT number,
       o_newtprice    IN OUT number,
       rprice         IN OUT number,
       cost           IN OUT number
)
IS
       ototal number;
       not_serializable EXCEPTION;
       PRAGMA EXCEPTION_INIT(not_serializable,-8177);
BEGIN
  LOOP BEGIN

    select o_totalprice
       into o_tprice
       from orders
       where o_orderkey = o_key;

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

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

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

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

    update orders
       set o_totalprice = o_newtprice
       where o_orderkey = o_key;

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

    EXIT;

  EXCEPTION
    WHEN not_serializable THEN
      ROLLBACK;
  END;

  END LOOP;

END doatrans;
END;
/

exit;
```

E.5 atranspl.c

```
/* Copyright (c) Oracle Corporation 2001. 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      02/22/01 - enlarge timing array
   mpoess      01/04/01 - Creation

*/

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

#include "atranspl.h"

/* Declare error handling functions */

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

/* declarations for ORDERS */

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

/* declarations for LINEITEM */

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

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

sb2 l_npricei;

/* other declarations */

int delta = 0;
double rprice;
double cost;

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

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

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

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

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

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

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

time_t curr_time;         /* Current Time                       */

/* OCI handles */

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

```
/* OCI bind handles */

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

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

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

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

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

char sqlstmt[1024];

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

void usage()
{

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

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

}


void ACIDexit() {

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

}



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

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

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

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

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

  ACIDexit();

  exit(1);
}

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

  int i;
```

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

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

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

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

  proc_no = atoi(argv[1]);

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

  num_streams = atoi(argv[2]);

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

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

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

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

  /* Process  optional parameters */

  argc -= 4;
  argv += 4;

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

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

  /* Initialize the cursors etc. */

  (void) ACIDinit();

  /* sleep for some time (triggering) */

  sleep(trig);

  /* start doing the ACID transactions */

  tr_start = gettime();

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

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

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

    /* Obtain l_key from l_key query */

    OCIsexec(tpcsvc,curi,errhp,1);

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

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

    /* Generate delta if necessary */

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

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

    curr_time = time(NULL);

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

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

    OCIsexec(tpcsvc,curr,errhp,1);

    curr_time = time(NULL);

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

```
    FPRTF1(outfile, "l_quantity:      %d\n", (int) l_quan);
    FPRTF1(outfile, "o_totalprice:    %.2f\n\n", o_tprice);
    }

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

    /* Shall we commit? */

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

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

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

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

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

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

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

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

  tr_end = gettime();

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

  /*  Disconnect from ORACLE. */

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

  ACIDexit();

  exit(0);
}


void ACIDinit()
{

  /* run random seed */

  srand48(getpid());

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

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

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

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

  /* get username and password */

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

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

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

OCIaset(tpcusr,OCI_HTYPE_SESSION,lname,strlen(lname),OCI_ATT
R_USERNAME,
        errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI_A
TTR_PASSWORD,
        errhp);

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


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

  /* Make session serializable */

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

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


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

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



OCIbbname(curi,&l_keyi_bp,errhp,":l_key",ADR(l_key),SIZ(l_key),SQL
T_INT);

OCIbbname(curi,&o_keyi_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),S
QLT_INT);

#endif /* NOLKEY */

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


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

  /* bind variables */


OCIbbname(curr,l_key_bp,errhp,":l_key",ADR(l_key),SIZ(l_key),SQLT_
INT);

OCIbbname(curr,o_key_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),SQ
LT_INT);

OCIbbname(curr,delta_bp,errhp,":delta",ADR(delta),SIZ(delta),SQLT_I
NT);

OCIbbname(curr,l_pkey_bp,errhp,":l_pkey",ADR(l_pkey),SIZ(l_pkey),S
QLT_INT);

OCIbbname(curr,l_skey_bp,errhp,":l_skey",ADR(l_skey),SIZ(l_skey),S
QLT_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_F
LT);

OCIbbname(curr,l_disc_bp,errhp,":l_disc",ADR(l_disc),SIZ(l_disc),SQL
T_FLT);

OCIbbname(curr,l_eprice_bp,errhp,":l_eprice",ADR(l_eprice),SIZ(l_epri
ce),
        SQLT_FLT);

OCIbbname(curr,l_neweprice_bp,errhp,":l_neweprice",ADR(l_newepric
e),
        SIZ(l_neweprice),SQLT_FLT);


OCIbbname(curr,o_tprice_bp,errhp,":o_tprice",ADR(o_tprice),SIZ(o_tpr
ice),
        SQLT_FLT);

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

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

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

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

  /* bind variables */


OCIbbname(cure1,l_neweprice1_bp,errhp,":l_neweprice",ADR(l_newe
price),
        SIZ(l_neweprice),SQLT_FLT);

OCIbbname(cure1,l_newquan1_bp,errhp,":l_newquan",ADR(l_newqua
n),
        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),SQ
LT_INT);


OCIbbname(cure2,o_newtprice2_bp,errhp,":o_newtprice",ADR(o_newt
price),
        SIZ(o_newtprice),SQLT_FLT);

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

}
```

E.6 atranspl.h

```
/* Copyright (c) Oracle Corporation 2001. All Rights Reserved. */

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

   DESCRIPTION

   MODIFIED   (MM/DD/YY)
   mpoess      01/04/01 - Creation

*/
#ifndef ATRANSPL_H

#define ATRANSPL_H

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

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

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

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

extern int errno;

#ifndef NULL
#define NULL 0
#endif

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

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#ifndef ub1
#define ub1 unsigned char
#endif

#define UNAME_LEN 64
#define WRITE_BUF_LEN 1024

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

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

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

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

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

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

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

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

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

#define OCIcom(svcp,errh) \
```

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

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

#define ISOTXT "alter session set isolation_level = serializable"

#define SQLTXT1 "BEGIN SELECT MAX(l_linenumber) INTO :l_key
FROM lineitem \
WHERE l_orderkey = :o_key; END;"

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

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

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

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

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

#endif /* ATRANSPL_H */
```

E.7 ckpt.sh

```
#!/bin/ksh
#
# $Header: ckpt.sh 08-aug-99.17:32:22 mpoess Exp $
#
# ckpt.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      ckpt.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
#
. $KIT_DIR/env
sqlplus -s /NOLOG << !

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

E.8 cnt_hist.sql

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

E.9 consist.sh

```
#!/bin/ksh

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

. $KIT_DIR/env

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

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

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

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

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

usage() {

  echo ""
  echo "Usage: $0 [-n iter] [-s number of stream] [-p prog] [-u usr/pswd]
-h"
  echo ""
  echo "-n iter            : number of iterations, default is 100"
  echo "-s number of stream : number of streams, default is 2"
  echo "-p prog            : program to run, default is atranspl.ott"
```

```
  echo "-u usr/pswd         : user/password for database access, default
is tpcd/tpcd"
  echo "-t chkpt            : time after the start of ACID transaction to
perform the checkpoint"
  echo "                    default is 10 seconds"
  echo "-h                  : print this usage summary"
  exit 1;
}

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

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


if [ $ITER -lt 100 ]
then
echo "Error: Must at least run 100 iterations!"

echo "Exiting..."
exit 1
fi

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

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

i=0

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

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

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

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

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

i=0

while [ $i -lt $STREAM ]
do

   $PROG $i $STREAM 1 0 u${USER} i${KEY}${i} o${OUTFILE}${i} s1
&
   i=`expr $i + 1`
done

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

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

wait

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

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

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


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

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

done
```

E.10 consist.sql

```
Rem
Rem $Header: consist.sql 08-aug-99.16:59:17 mpoess Exp $
Rem
Rem consist.sql
Rem
Rem  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
Rem
Rem    NAME
Rem      consist.sql - <one-line expansion of the name>
Rem
Rem  DESCRIPTION
Rem       Verifies the consistency of TPC-D database using the
Rem       consistency condition.
Rem
Rem Usage:   sqlplus tpcd/tpcd @consist
Rem
Rem NOTE
```

```
Rem REQUIRES PACKAGES prvtotpt and dbmsotpt
rem
Rem    MODIFIED   (MM/DD/YY)
Rem    mpoess      08/08/99 - Creation
Rem    mpoess      08/08/99 - Created
Rem

set verify off
rem set termout on
rem set echo on


REM
REM Get today's date.
REM

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

set serverout on;

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


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

   diff := l_tprice - o_tprice;

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

END;
.
/


spool off
exit
```

E.11 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 commited $ITER transactions
```

E.12 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;
```

E.13 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=${NUM_STREAMS}
let STEM="$STEM + 1"
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=tpch/tpch
TRIG=1
HCNT=duracnta

# get history count

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

# perform the consistency

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

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

cat $ORACLE_HOME/rdbms/log/alert_qual.log >
${DURA_DIR)/alert_qual.log.end_dura 2>&1
```

E.14 iso.sh

```
#!/bin/ksh
#
# $Header: iso.sh 17-aug-99.15:44:51 mpoess Exp $
#
# iso.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      iso.sh
#
#    DESCRIPTION
#     This script triggers all 6 isolation tests. In addition,
#     it creates more readable formats of the isolation test output.
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/17/99 - Creation
#    mpoess      08/17/99 - Creation
#

for iso in iso1 iso2 iso3 iso4 iso5 iso6;do
        echo Running isolation test $iso
        ${iso}.sh
        echo Creating nicely formated output of ACID test $iso
        xiso.pl -o ${ACID_OUT}/${iso}
done
```

E.15 iso1.sh

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

. $KIT_DIR/env

# May need to change the following:
RSH=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; RMHOST="$1";;
  -h) usage; exit 0;;
  --) break;;
  esac
  shift;
done


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

# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

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

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

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

sleep 1

# start ACID transaction, Sleep for 60 second before COMMIT

$PROG 1 1 1 0 i$KEYFILE u$USER s60 >> $TXN1FILE &
#$PROG 1 1 1 0 i$KEYFILE u$USER s60 b0 >> $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 "`date`" >> $TXN2FILE
if [ "$RMHOST" != "" ]
then
echo "Starting ACID query on node $RMHOST" >> $TXN2FILE
${RSH} ${RMHOST}  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
```

E.16 iso2.sh

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

. $KIT_DIR/env

RSH=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; RMHOST="$1";;
  -h) usage; exit 0;;
  --) break;;
  esac
  shift;
done


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

OKEY=`cat $KEYFILE | awk '{print $1}'`
```

```
echo "o_key is "$OKEY

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

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

sleep 1

# start ACID transaction, Sleep for 30 second before ROLLBACK

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

# let's sleep 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 "`date`" >> $TXN2FILE
if [ "$RMHOST" != "" ]
then
echo "Starting ACID query on node $RMHOST" >> $TXN2FILE
${RSH} ${RMHOST}  sqlplus "$USER" @$ACID_DIR/isolarion/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
```

E.17 iso3.sh

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

. $KIT_DIR/env

# May need to change the following:
RSH=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; RMHOST="$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 >> $TXN1FILE &
#$PROG 1 2 1 0 i$KEYFILE u$USER s30 b0 >> $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 [ "$RMHOST" != "" ]
then
echo "Starting TXN2 on node $RMHOST" >> $TXN2FILE
```

```
${RSH} ${RMHOST} $PROG 2 2 1 1 i$KEYFILE u$USER s1 >>
$TXN2FILE &
else
$PROG 2 2 1 1 i$KEYFILE u$USER s1 >> $TXN2FILE &
#$PROG 2 2 1 1 i$KEYFILE u$USER s1 b1 >> $TXN2FILE &
fi

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

E.18 iso4.sh

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

. $KIT_DIR/env

# May need to change the following:
RSH=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; RMHOST="$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 >> $TXN1FILE &
#$PROG 1 2 0 0 i$KEYFILE u$USER s30 b0 >> $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 [ "$RMHOST" != "" ]
then
echo "Starting TXN2 on node $RMHOST" >> $TXN2FILE
$${RSH} ${RMHOST} $PROG 2 2 1 1 i$KEYFILE u$USER s1 >>
$TXN2FILE &
else
$PROG 2 2 1 1 i$KEYFILE u$USER s1 >> $TXN2FILE &
#$PROG 2 2 1 1 i$KEYFILE u$USER s1 b1 >> $TXN2FILE &
fi

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

E.19 iso5.sh

```
#!/bin/ksh
#
# $Header: iso5.sh 04-aug-99.09:21:45 mpoess Exp $
#
# iso5.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#   NAME
#     iso5.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#     Usage: iso5.sh [-u user/password] [-n remote_node] -h
#     Options: See usage below
```

```
#  NOTES
#       For a cross node isolation test, assume the local node is
#       one of the participating nodes.  The other node can be
#       specified by the -n option.
#       You need to set the environment variable TPCD_KIT_DIR
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/04/99 - Creation
#    mpoess      08/04/99 - Creation
#

. $KIT_DIR/env

# May need to change the following:
RSH=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; RMHOST="$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 [ "$RMHOST" != "" ]
then
echo "Starting PARTSUPP query on node $RMHOST" >> $TXN2FILE
${RSH} ${RMHOST} sqlplus $USER @$ACID_DIR/isolation/a_query2
${PSKEY} >> $TXN2FILE &

else
sqlplus $USER @$ACID_DIR/isolation/a_query2 ${PSKEY} >>
$TXN2FILE &
fi

wait

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

E.20 iso6.sh

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

. $KIT_DIR/env
```

```
# May need to change the following:
RSH=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; RMHOST="$1";;
  -h) usage; exit 0;;
  --) break;;
  esac
  shift;
done


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

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

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

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

sleep 1

# start Query 1, use 0 as the delta

echo "Running Query 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 [ "$RMHOST" != "" ]
then
echo "Starting ACID transaction on node $RMHOST" >> $TXN2FILE
${RSH}  ${RMHOST} $PROG 1 1 1 0 i$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
```

E.21 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;
OCIStmt *curi;

OCIBind *l_key_bp;
OCIBind *o_key_bp;

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

char sqlstmt[1024];


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


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

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

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

      (void) OCIErrorGet(errhp,1,NULL,(sb4 *) &errcode,(text *)msg,
                         2048,OCI_HTYPE_ERROR);
    else
      (void) OCIErrorGet(errhp,1,NULL,(sb4 *) &errcode,(text *)msg,
                         2048,OCI_HTYPE_ENV);
    fprintf(stderr,"%s\n",msg);
    break;
  }
  /* Rollback just in case */

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

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

  ACIDexit();

  exit(1);
}


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

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

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

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

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

  argc -= 2;
  argv += 2;

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

  ACIDinit();

  /* initialize array for random numbers */

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

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

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

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

    /* Obtain l_key from l_key query */

    OCIsexec(tpcsvc,curi,errhp,1);

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

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

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

  ACIDexit();
  free(res);

}


void usage() {

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

void ACIDinit()
{

  /* run random seed */

  srand48(getpid());

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

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

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

  /* get username and password */

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

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


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

OCIaset(tpcusr,OCI_HTYPE_SESSION,lname,strlen(lname),OCI_ATT
R_USERNAME,
          errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI_A
TTR_PASSWORD,
          errhp);

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


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

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

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


OCIbbname(curi,l_key_bp,errhp,":l_key",ADR(l_key),SIZ(l_key),SQLT_
INT);

OCIbbname(curi,o_key_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),SQ
LT_INT);
}
```

E.22 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");
}
```

E.23 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 tpch/tpch @sample $j
i=`expr $i + 1`
done

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

E.24 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;
```

E.25 q1.sql

```
Rem

Rem $Header: template.sql 06-feb-96.13:23:14 mpoess   Exp $
Rem
Rem q1.sql
Rem
Rem  Copyright (c) Oracle Corporation 2001. All Rights Reserved.
Rem
Rem    NAME
Rem      q1.sql - <one-line expansion of the name>
Rem
Rem    DESCRIPTION
Rem      used in isolation test 6
Rem
Rem    NOTES
Rem      <other useful comments, qualifications, etc.>
Rem
Rem    MODIFIED   (MM/DD/YY)
Rem    mpoess      02/13/01 - Created
Rem

set serverout on;

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

```
from dual;

select
        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')  - 0
group by
        l_returnflag,
        l_linestatus
order by
        l_returnflag,
        l_linestatus;

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

exit;
```

E.26 run_acid.sh

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

#    mpoess      08/08/99 - Creation
#    mpoess      08/08/99 - Creation
#

. $KIT_DIR/env

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

usage() {

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

#ITER=100
ITER=200
STEM=$NUM_STREAMS
let STEM="$STEM + 1"
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=tpch/tpch
TRIG=1
HCNT=duracntb

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

# get all the options

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


echo "Starting ACID run..."

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

# Get history count before the run

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

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

wait
# perform the consistency

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

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

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

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

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

done

wait

echo "ACID run completed"
```

E.27 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 $DATABASE_USER @atrans
sqlplus $DATABASE_USER @d_hist
```

Appendix - F Seed and Input Parameters

F.1 Seed

0907014051

F.2 qp2.0

```
14    1993-12-01
2     13     BRASS   AMERICA
9     white
20    thistle 1997-01-01     ROMANIA
6     1997-01-01     0.09    25
17    Brand#14       JUMBO BOX
18    314
8     RUSSIA  EUROPE  PROMO BRUSHED STEEL
21    UNITED KINGDOM
13    special packages
3     AUTOMOBILE     1995-03-08
22    22     17     19     33     18     29     11
16    Brand#41       LARGE POLISHED  16     42     48     41     19
17    8      2
4     1994-04-01
11    ROMANIA 0.0000000333
15    1993-11-01
1     65
10    1994-10-01
19    Brand#45       Brand#54       Brand#21       2      11     20
5     ASIA    1997-01-01
7     UNITED KINGDOM  RUSSIA
12    FOB     REG AIR 1993-01-01
```

F.3 qp2.1

```
21    MOZAMBIQUE
3     HOUSEHOLD      1995-03-25
18    315
5     MIDDLE EAST    1997-01-01
11    GERMANY 0.0000000333
7     MOROCCO KENYA
6     1997-01-01     0.06    25
20    ghost   1996-01-01     INDONESIA
17    Brand#15       JUMBO JAR
12    MAIL    REG AIR 1994-01-01
16    Brand#21       PROMO BRUSHED   35     6      28     19     43
18    41     1
15    1996-06-01
13    special packages
10    1993-08-01
2     50     TIN     EUROPE
8     KENYA   AFRICA  PROMO PLATED STEEL
14    1994-03-01
19    Brand#42       Brand#42       Brand#21       7      12     27
9     tan
22    12     34     27     26     11     33     10
1     73
4     1996-11-01
```

F.4 qp2.2

```
6     1993-01-01     0.04    24
17    Brand#12       JUMBO CAN
14    1994-06-01
16    Brand#11       MEDIUM BURNISHED        45     1      16     48
32    44     47     15
19    Brand#44       Brand#25       Brand#15       3      13     23
10    1994-05-01
9     sky
2     38     COPPER  AMERICA
15    1994-03-01
8     FRANCE  EUROPE  PROMO ANODIZED STEEL
5     AFRICA  1993-01-01
22    33     30     32     15     19     26     27
12    TRUCK   REG AIR 1994-01-01
7     GERMANY FRANCE
13    special requests
18    313
1     81
4     1994-08-01
20    rose    1994-01-01     UNITED KINGDOM
3     AUTOMOBILE     1995-03-10
11    SAUDI ARABIA    0.0000000333
21    INDIA
```

F.5 qp2.3

```
8     UNITED KINGDOM  EUROPE  ECONOMY POLISHED STEEL
5     AMERICA 1993-01-01
4     1997-02-01
6     1993-01-01     0.09    25
17    Brand#14       WRAP BOX
7     UNITED STATES   UNITED KINGDOM
1     89
18    315
22    23     11     10     13     24     22     28
14    1994-10-01
9     royal
10    1993-02-01
15    1996-10-01
11    INDIA   0.0000000333
20    coral   1993-01-01     JORDAN
2     26     STEEL   MIDDLE EAST
21    ALGERIA
19    Brand#51       Brand#13       Brand#14       8      14     30
13    special requests
16    Brand#41       ECONOMY PLATED  3      34     42     5      32
17    22     20
12    RAIL    REG AIR 1994-01-01
3     FURNITURE      1995-03-27
```

F.6 qp2.4

```
5     ASIA    1993-01-01
21    PERU
14    1995-01-01
19    Brand#53       Brand#41       Brand#13       3      15     26
15    1994-06-01
17    Brand#11       WRAP JAR
12    REG AIR AIR     1994-01-01
6     1993-01-01     0.07    25
4     1994-11-01
9     powder
8     MOROCCO AFRICA  ECONOMY BURNISHED COPPER
16    Brand#21       STANDARD BRUSHED        5      37     15     7
45    2      47     25
11    VIETNAM 0.0000000333
2     14     BRASS   AMERICA
10    1993-11-01
18    312
1     97
13    special requests
7     MOZAMBIQUE     MOROCCO
22    20     21     15     16     26     25     23
3     MACHINERY      1995-03-12
20    navajo  1996-01-01     CANADA
```

F.7 qp2.5

```
21    INDONESIA
15    1997-01-01
4     1997-06-01
```

6 1993-01-01 0.04 24
7 INDIA GERMANY
16 Brand#11 LARGE ANODIZED 2 19 3 30 14
33 22 29
19 Brand#51 Brand#24 Brand#53 8 16 23
18 314
14 1995-04-01
22 28 32 19 29 24 30 11
11 INDONESIA 0.0000000333
13 special requests
3 FURNITURE 1995-03-29
1 105
2 1 NICKEL MIDDLE EAST
5 EUROPE 1993-01-01
8 GERMANY EUROPE LARGE BRUSHED COPPER
20 antique 1994-01-01 CHINA
12 SHIP AIR 1995-01-01
17 Brand#13 WRAP CAN
10 1994-08-01
9 pale

F.8 qp2.6

10 1993-06-01
3 MACHINERY 1995-03-14
15 1994-10-01
13 special requests
6 1994-01-01 0.02 25
8 UNITED STATES AMERICA LARGE PLATED COPPER
9 moccasin
7 ALGERIA UNITED STATES
4 1995-03-01
11 RUSSIA 0.0000000333
22 17 24 20 21 26 34 28
18 315
12 FOB AIR 1995-01-01
1 113
5 MIDDLE EAST 1994-01-01
16 Brand#41 PROMO PLATED 2 22 4 18 49
11 3 10
2 39 COPPER ASIA
14 1995-07-01
19 Brand#13 Brand#12 Brand#52 4 17 30
20 khaki 1993-01-01 GERMANY
17 Brand#15 SM BOX
21 ARGENTINA

F.9 qp2.7

18 313
8 MOZAMBIQUE AFRICA LARGE ANODIZED COPPER
20 sky 1996-01-01 RUSSIA
21 ROMANIA
2 27 STEEL MIDDLE EAST
4 1997-10-01
22 31 29 14 17 21 24 26
17 Brand#12 SM JAR
1 60
11 IRAN 0.0000000333
9 maroon
19 Brand#15 Brand#45 Brand#41 9 18 26
3 BUILDING 1995-03-31
13 pending accounts
5 AFRICA 1994-01-01
7 MOZAMBIQUE JORDAN
10 1994-03-01
16 Brand#21 SMALL POLISHED 11 43 44 46 40
8 42 18
6 1994-01-01 0.07 25
14 1995-10-01
15 1997-05-01
12 MAIL AIR 1995-01-01

F.10 qp2.8

19 Brand#12 Brand#33 Brand#45 4 19 22
1 68
15 1995-01-01
17 Brand#14 SM CAN
5 AMERICA 1994-01-01
8 INDIA ASIA MEDIUM POLISHED COPPER
9 lawn
12 TRUCK RAIL 1996-01-01
14 1996-02-01
7 INDIA IRAQ
4 1995-07-01
3 MACHINERY 1995-03-16
20 dodger 1995-01-01 JAPAN
16 Brand#11 ECONOMY ANODIZED 26 14 4 32
47 48 29 43
6 1994-01-01 0.04 24
22 10 28 20 25 17 18 29
10 1994-12-01
13 pending accounts
2 15 BRASS ASIA
21 IRAQ
18 314
11 UNITED KINGDOM 0.0000000333

Appendix - G Pricing

G.1 Oracle Pricing

From: MaryBeth Pierantoni <mary.beth.pierantoni@oracle.com>
Sent: 09/13/2005 06:23 PM
To: Tony Petrossian
Subject: Oracle Pricing

Product	Price	Qty	Extended Price
Oracle Database 10g Enterprise Edition Release 2, Named User Plus for 3 years	10,000	48*	480,000
Partitioning, Named User Plus for 3 years	2,500	48*	120,000
Oracle Database Server Support Package for 3 years	2,000	3	6,000
Oracle Mandatory E-Business Discount			<121,200>
Total			**484,800**

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

* 48 = 0.75 * 64. Explanation: For the purposes of counting the number of processors which require licensing, a multicore chip with "n" cores shall be determined by multiplying "n" cores by a factor of .75.

Regards,
MaryBeth Pierantoni

G.2 IBM Pricing

International Business Machines Corporation

3200 Windy Hill Road
Atlanta, Georgia 30339

September 15, 2005

Tony Petrossian
IBM Corporation

Dear Tony,
The requested quote for the p5 595 TPC-H benchmark is below.

Description	Qty	List Price	Extended Purchase	3yr Maint 24x7
P5 Server Model 595	1	$92,000.00	$92,000.00	$63,000.00
RIO-2 (Remote I/O-2) Cable, 3.5M	8	$660.00	$5,280.00	
RIO-2(Remote I/O-2)Cbl, 8.0M	24	$800.00	$19,200.00	
36.4 GB 15,000 RPM Ultra320 SCSI Disk Drive Assembly	4	$2,125.00	$8,500.00	
Service Shelf Tool Kit	1	N/C	N/C	
Software Preinstall	1	N/C	N/C	
IBM 10/100/1000 Base-TX Ethernet PCI-X Adapter	1	$1,280.00	$1,280.00	
PCI-X Dual Ultra320 SCSI by Adaptec, Inc.	1	$790.00	$790.00	
2 Gigabit Fibre Channel PCI-X Adapter	96	$2,720.00	$261,120.00	
Expansion Rack, Powered	1	$60,000.00	$60,000.00	$3,960.00
I/O Drawer, 20 Slots, 8 Disk Bays	8	$30,000.00	$240,000.00	$32,640.00
I/O Drw.Cbl.Grp, Prim.Rck/5U	1	$400.00	$400.00	
I/O Drw.Cbl.Grp, Prim.Rck/1U	1	$400.00	$400.00	
Bulk Power Regulator	10	$4,000.00	$40,000.00	
Slim Line Doors	2	$6,000.00	$12,000.00	
Ethernet Cable, 15M, Hardware Management Console to System Unit	3	$31.00	$93.00	
Bulk Power Controller Assembly	4	$4,000.00	$16,000.00	
Cooling Group, 2-4 Processor Books	1	$4,000.00	$4,000.00	
DC Power Converter, Processor Book	12	$6,000.00	$72,000.00	
Processor Clock Card, Programmable	2	$575.00	$1,150.00	
System Service Processor	2	$3,500.00	$7,000.00	
Multiplexer Card	4	$2,200.00	$8,800.00	
16-Way POWER5 Turbo CUoD Processor Book, 0-Way Active 1.9 GHz	4	$127,000.00	$508,000.00	$68,640.00
Activation, #7813 CUoD Processor Book, One Processor	64	$32,000.00	$2,048,000.00	$506,880.00
Remote I/O-2 (RIO-2) Loop Adapter, Two Port	16	$3,400.00	$54,400.00	
Pwr.Cbl.Grp, CEC Primary Fans	1	$650.00	$650.00	
Pwr.Cbl.Grp, 1st CEC Book	1	$650.00	$650.00	
Pwr.Cbl.Grp, 2nd CEC Book	1	$650.00	$650.00	
Pwr.Cbl.Grp, 3rd CEC Book	1	$650.00	$650.00	
Pwr.Cbl.Grp, 4th CEC Book	1	$650.00	$650.00	
Pwr.Cbl.Grp, 7807 Cooling Grp.	1	$650.00	$650.00	
Bulk Power Distribution Assembly	6	$2,500.00	$15,000.00	
Power Cables, 4x, 01U	1	$650.00	$650.00	
Power Cables, 4x, 05U	1	$650.00	$650.00	
Power Cables, 4x, 09U	1	$650.00	$650.00	

Power Cables, 4x, 19U	1	$400.00	$400.00	
Power Cables, 4x, 23U	1	$400.00	$400.00	
Power Cables, 4x, 27U	1	$400.00	$400.00	
Advanced POWER Virtualization	64	N/C	N/C	
256GB DDR1 Memory (32 X 8GB)	1	$786,432.00	$786,432.00	
Line Cord, 6AWG/Type W, 14ft, IEC309 60A Plug	4	$2,000.00	$8,000.00	
Language Group Specify - US English	1	N/C	N/C	
		Subtotal	**$4,276,895.00**	**$675,120.00**
IBM DVD-RAM Drive	1	$1,400.00	$1,400.00	$360.00
English Manuals	1	N/C	N/C	
2.5m VHDCI/HD68 SCSI Cable	1	N/C	N/C	
9 FT 125V 15A Power Cord	1	N/C	N/C	
		Subtotal	**$1,400.00**	**$360.00**
HMC 1:7310-C04 Desktop Hardw.Mgmt.Console	1	$1,830.00	$1,830.00	$768.00
7310-C04 24x7x4 WSU	1	$192.00	N/A	$192.00
IBM ThinkVision C170 17-inch Color Monitor	1	$250.00	$250.00	
10/100 Mbps Ethernet PCI Adapter II	1	$412.00	$412.00	
Power Cord (6-foot), To Wall (125V, 15A), Plug Type #4	1	$14.00	$14.00	
Ethernet Cable, 6M, Hardware Management Console to System Unit	1	$12.00	$12.00	
Quiet Touch Keyboard - USB, Business Black, US English, #103P	1	$83.00	$83.00	
Mouse - Business Black with Keyboard Attachment Cable	1	$62.00	$62.00	
Language Group Specify - US English	1	N/C	N/C	
		Subtotal	**$2,663.00**	**$960.00**
DS4000 EXP710 Storage Expansion	96	$6,000.00	$576,000.00	N/A
1740-710 24x7x4 WSU	96	$760.00	N/A	$72,960.00
(19K1271) 2Gb Fibre Channel Short Wave GBIC	192	$499.00	$95,808.00	
(06P5772) 2Gb FC, 36.4GB/15K Drive	1152	$1,115.00	$1,284,480.00	
(19K1247) 1m 50u Fiber Optic Cable (LC-LC)	192	$79.00	$15,168.00	
Attach to the DS4500 (1742-90U)	96	N/C	N/C	
Field Integrate FC EXP 710	96	N/C	N/C	
(6952300) Power Cord 125V/10A, (Group 1)	96	N/C	N/C	
		Subtotal	**$1,971,456.00**	**$72,960.00**
DS4500 Midrange Disk System	24	$49,900.00	$1,197,600.00	N/A
1742-90U 24x7x4 WSU	24	$1,087.00	N/A	$26,088.00
(19K1271) 2Gb Fibre Channel Short Wave GBIC	288	$499.00	$143,712.00	
(19K1269) DS4000 2 Gbps miniHUB (LC)	96	$899.00	$86,304.00	
(19K1249) Fiber Cable 25m Multimode (LC-LC)	96	$189.00	$18,144.00	
(24P7981) DS4500 AIX Host Kit	1	$4,000.00	$4,000.00	
Field Integrate DS4500	24	N/C	N/C	
(6952300) Power Cord 125V/10A, (Group 1)	24	N/C	N/C	
		Subtotal	**$1,449,760.00**	**$26,088.00**
IBM S2 42U Standard Rack	10		$14,890.00	N/A
3-Year 24x7x4 Support ServicePac for Rack	10		NA	$3,000.00
		Subtotal	**$14,890.00**	**$3,000.00**
System Software	1	N/C	N/C	
MEDIA 5765-G03 AIX V5.3	1	N/C	N/C	
Expansion pack	1	N/C	N/C	
Aix 5.3 Update CD	1	N/C	N/C	
Microcode Upd Files and Disc Tool CD	1	N/C	N/C	
Virtual I/O Server	1	N/C	N/C	
CD-ROM Process Charge	1	$50.00	$50.00	
Preinstall 64-bit Kernel	1	N/C	N/C	
Partion Load Manager	1	N/C	N/C	

English Language	1	N/C	N/C	
CD-ROM	1	N/C	N/C	
Suppress Single Pallet Delivery	1	N/C	N/C	
Preinstall	1	N/C	N/C	
English U/L SBCS Secondary Language	1	N/C	N/C	
		Subtotal	**$50.00**	**N/A**
AIX V5.3	1	N/C	N/C	
Per Processor H5 AIX V5.3	64	$2,495.00	$159,680.00	
		Subtotal	**$159,680.00**	**N/A**
Partition Load Manager	1	N/C	N/C	
Per Processor H5 Partition Load Mgr	64	N/C	N/C	
		Subtotal	**N/C**	**N/A**
Virtual I/O Server	1	N/C	N/C	
Per Processor H5 Virtual I/O Server	64	N/C	N/C	
		Subtotal	**N/C**	**N/A**
Partition Load Manager SW Maint: 3 year	1	N/C	N/C	
5773-PL3 SW MAINT 3Y Reg H5	64	$35.00	$2,240.00	$2,240.00
Per processor 3 Yr SW Maint 24x7 Support H5	64	$14.00	$896.00	$896.00
		Subtotal	**N/A**	**$3,136.00**
Software Maintenance for AIX, 3 Year	1	N/C	N/C	N/C
H5 3 Yr SWMA for AIX per Processor Reg/Ren	64	$2,836.00	$181,504.00	$181,504.00
H5 3 yr Services 7x24 Support per Processor Reg/Ren	64	$732.00	$46,848.00	$46,848.00
		Subtotal	**N/A**	**$228,352.00**
Virtual I/O Server SW Maintenance: 3 Yr	1	N/C	N/C	N/C
Per Processor H5 VIO 3 Yr Maintenance	64	$155.00	$9,920.00	$9,920.00
Per Processor H5 VIO 3 Yr Maint 24x7 Support	64	$64.00	$4,096.00	$4,096.00
		Subtotal	**N/A**	**$14,016.00**
Initial Software Support 3 Year	1	N/C	N/C	N/C
Per Processor Software Support 3 Year	1	$675.00	$675.00	$675.00
Per Processor 24x7 Software Support 3 Year	1	$236.00	$236.00	$236.00
Agreement for MCRSA	1	N/C	N/C	N/C
		Subtotal	**N/A**	**$911.00**
C for AIX user Lic+SW maint 12 MO	1	$515.00	$515.00	N/A
C for AIX user annual SW maint renewal	2	NA	NA	$206.00
		Subtotal	**$515.00**	**$206.00**
		Total	**$7,877,309.00**	**$1,025,109.00**
		IBM HW/SW Configuration Total Discount	**$4,028,344.00**	
		3yr Total Cost for IBM Configuration	**$4,874,074.00**	

For additional information, please contact me directly at 770-835-6612.

Alycia E. Waller
Technical Sales Specialist - eServer pSeries
awaller@us.ibm.com