HP Integrity Superdome Enterprise Server
using
HP-UX 11.i V2 64-bit
and
Oracle Database 10g Enterprise Edition with Partitioning

TPC Benchmark™ H
Full Disclosure Report

Second Edition

December 24, 2004

Second Edition - December 24, 2004

Hewlett-Packard Company, the sponsor of this benchmark test, believes that the information in this document is accurate as of the publication date. The information in this document is subject to change without notice. The sponsors assume no responsibility for any errors that may appear in this document. The pricing information in this document is believed to accurately reflect the current prices as of the publication date. However, the sponsors provide no warranty of the pricing information in this document.

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

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

Printed in U.S.A., December 24, 2004.

Overview

This report documents the methodology and results of the TPC Benchmark™ H test conducted on the HP Integrity Superdome Enterprise Server, in conformance with the requirements of the TPC Benchmark™ H Standard Specification, Revision 2.1.0. The operating system used for the benchmark was HP-UX 11.i V2 64-bit; the DBMS was Oracle 10g.

Standard and Executive Summary Statements

The pages following this preface contain the Executive Summary and Numerical Quantities Summary of the benchmark results.

Auditor

The benchmark configuration, environment and methodology used to produce and validate the test results and the pricing model used to calculate the cost per QphH was audited by Francois Raab, InfoSizing, to verify compliance with the relevant TPC specifications.

TPC Benchmark H Overview

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

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

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

- Give answers to real-world business questions;
- Simulate generated ad-hoc queries(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 updates 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 updates which batch together a number of modifications impacting some part of the decision support database;
- Due to the world-wide nature of the business data stored in the TPC-H database, the queries and the updates may be executed against the database at any time, especially in relation to each other. In addition, this mix of queries and updates is subject to specific ACIDity requirements, since queries and updates 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 updates.

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 GB. Compliant benchmark implementations may also use one of the larger permissible database populations (e.g. 10000 GB), as defined in Clause 4.1.3.

The performance metrics reported by TPC-H measure multiple aspects of the capability of the system to process queries. The TPC-H metric at the selected size (QphH@Size) is the performance metric. 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 5.4.7). 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-D 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.

General Implementation Guidelines

The purpose of TPC benchmarks is to provide relevant, objective performance data to industry users. To achieve that purpose, TPC benchmark specifications require that benchmark tests be implemented with systems, products, technologies and pricing that:

- Are generally available to users;
- Are relevant to the market segment that the individual TPC benchmark models or represents (e.g. TPC-H models and represents complex, high data volume, decision support environments);
- Would plausibly be implemented by a significant number of users in the market segment the benchmark models or represents.

Hewlett-Packard Company does not warrant or represent that a user can or will achieve performance similar to the benchmark results contained in this report. No warranty of system performance or price/performance is expressed or implied by this report

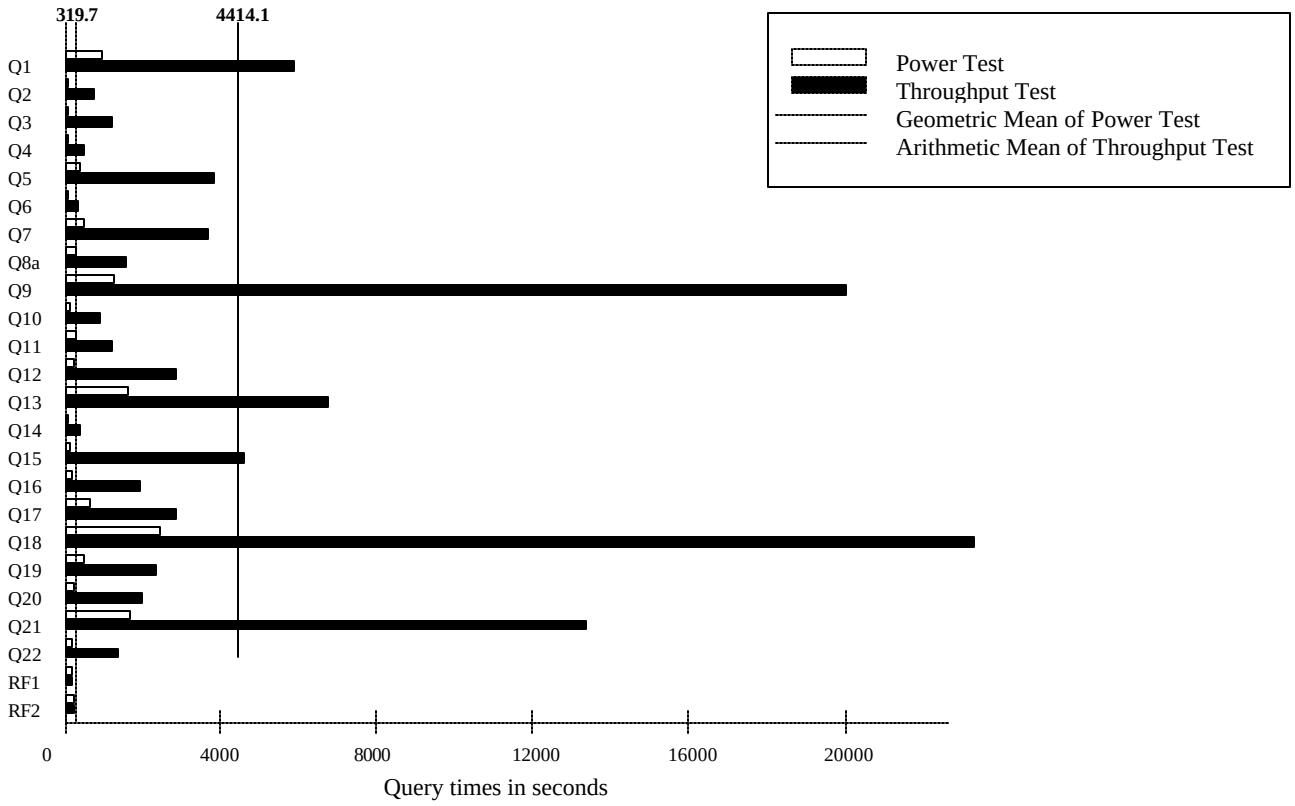
HP Integrity Superdome Enterprise Server

TPC-H Rev 2.1.0

Report Date: December 24, 2004

Total System Cost	Composite Query per Hour Metric	Price/Performance
$13,912,046	**86,282.7 QphH@10000GB**	**$161 QphH@10000GB**

Database Size	Database Manager	Operating System	Other Software	Availability Date
10000 GB*	**Oracle Database 10g Enterprise Edition with Partitioning**	**HP-UX 11.i V2 64-bit**	**None**	**April 6, 2005**



Database Load Time = 07:19:00	Load Includes Backup: N	Total Data Storage/Database Size = 10.53
RAID (Base Tables Only): N	RAID (Base Tables and Auxiliary Data Structures): N	RAID (All): Y

System Configuration

Processors:	128 Itanium2 1.5GHz, 6MB L3 Cache
Memory:	512 GB
Disk Drives:	2 HP Surestore Disk System 2100 with 8 - 36GB 1-36GB LP 10K LVD SE U320 HDD disks and 22 HP Disk Array XP128 (with a total of 1452 73GB 10K RPM disks)
Total Disk Storage	105264GB (In this calculation one GB is defined as 1024*1024*1024 bytes)
Lan Controllers	2 PCI 1000BT Lan Adapter

*Database Size includes only raw data (e.g. no temp, index, redundant storage space, etc.)

HP Integrity Superdome Enterprise Server

TPC-H Rev 2.1.0

Report Date: December 24, 2004

Description	Part Number	Source	Reference Price	Qty	Extended Price	3 yr. Maint. Price
Server Hardware						
Superdome left chassis	A5201A, Opt. 429	1	205,840	2	411.680	
Superdome right chassis	A5202A, Opt. 429	1	218,435	2	436.870	
IPF Superdome Cell Board (sx1000)	A6866A	1	16,000	32	512.000	
3 Year Svc & Support Price (Hardware and Software)						$1,622,611
4GB SDRAM (4x1GB DIMMS)	A6863A	1	13,898	128	1.778.944	
PCI-x I/O chassis	A6864A	1	16,805	32	537.760	
Core I/O Card	A6865A	1	1,045	2	2.090	
CPU Itanium 2, 1.5GHz w/6MB iL 3 cache (2 CPUs)	A6924A	1	38,000	64	2.432.000	
PCI 1000BT Lan Adapter	A6847A, Opt. 0D1	1	1,325	2	2.650	
HP PCI-X 2-port 4X Fabric (HCA) Adapter	AB286A	1	2,835	8	22.680	
HP 24-port 4X Fabric Copper Switch	AB399A	1	11,800	1	11.800	
HP 5m 4x Fabric Copper Cable	AB346A	1	275	4	1.100	
I/O chassis enclosure for PCI chassis	A5862A	1	25,725	8	205.800	
Graphite I/O expansion power subsystem	A5861D	1	34,860	4	139.440	
PCI 2GB Fibre Channel Adapter	A6795A	1	2,195	176	386.320	
PCI Ultra160 SCSI Adapter	A6828A	1	1,049	2	2.098	
HP Surestore Disk System 2100	A5675A	1	699	2	1.398	
1-36GB LP 10K LVD SE U320 HDD	A6571A	1	651	8	5.208	
HP Rack System/E, 41U	A4902D	1	1,910	2	3.820	
Modular Power Dist Unit for std racks	A5137AZ	1	145	2	290	
200-240 volts North America	A5137AZ	1	94	2	188	
TA5300 Enclosure for DAT tape	C7508AZ	1	1,045	2	2.090	
DDS 4 tape	C7497B	1	1,049	2	2.098	
DVD Rom drive	C7499A	1	515	2	1.030	
SCSI Terminator LVD/SE HDTS68 Multimedia	C2364A	1	100	2	200	
HP Tape Array PSU/Fan Kit	C7496A	1	319	2	638	
SCSI Cable 10m VHDTS68/DHTS68 M/M Multimd	C2363B	1	335	4	1.340	
SCSI Cable 0.5m HDTS68 M/M Multimedia	C2978B	1	99	2	198	
SX1000 Superdome SMS, rack	A9802A	1	5,140	2	10.280	
1U Rackmt Display/Keyboard/Mouse	AB243AZ	1	3,046	1	3.046	
				Subtotal	**6,915,056**	**1,622,611**
Server Software						
Oracle Database 10g Enterprise Edition, Named User Plus for 3 years		2	10,000	128	1.280.000	
Real Application Clusters, Named User Plus for 3 years		2	5,000	128	640.000	
Partitioning, Named User Plus for 3 years		2	2,500	128	320.000	
Oracle Database Server Support Package for 3 years:		2	12,000	1		12,000
HPUX 11i. V2 Foundation Operating Environment	B9429AC	1	2,370	128	303.360	
HPUX Fndn OE Media	B9106AA, Opt OD1	1	199	2	398	
				Subtotal	**2,543,758**	**12,000**
Storage						
HP Disk Array XP128 SSP Solution	A7875A	1	0	22	0	
XP128 Disk Control Frame	A7876A	1	55,880	22	1.229.360	
HP Proactive 24 Support Upgrade - 3 yrs		1				936,859
XP1024/128 8 Port 1-2GB/sec FC/CA CHIP Pair	A7912B	1	52,490	44	2.309.560	
XP1024/128 2GB Cache Memory Module	A7918A	1	23,170	154	3.568.180	
XP1024/128 512MB Shared Memory Module	A7921A	1	7,170	110	788.700	
XP1024/128 Array Control Processor (ACP)	A7922A	1	58,845	44	2.589.180	
XP128 Disk Path Expansion Kit	A7894A	1	19,190	22	422.180	
XP128 73GB 10k rpm, FC Array Group(4 disks)	A7929A	1	10,417	352	3.666.784	
XP128 73GB 10K rpm, FC Disk Drives	A7929S	1	2,590	44	113.960	
HP ProCurve Switch 2124	A7929S	1	279	1	279	
LUN Conf/Sec Mgr XP 1TB LTU (up to 1TB)	T1614AA	1	13,666	22	300.652	
LUN Conf/Sec Mgr XP 1TB LTU (2-6TB)	T1614AB	1	6,309	88	555.192	
Command View XP LTU	B9357AJ	1	24,690	2	49.380	
16m Fibre Channel Cables	C7525A	1	260	176	45.760	
				Subtotal	**15,639,167**	**936,859**
				Total	**25,097,981**	**2,571,470**
Oracle Mandatory E-Business Discount on (Licenses and Support)					(563,000)	
Large Configuration Discount and Support Prepayment*					(11,771,146)	(1,423,259)
				Grand Total	**12,763,835**	**1,148,211**

Source: 1=HP,
2=Oracle (Pricing Contact: MaryBeth Pierantoni;
email: mary.beth.pierantori@oracle.com; phone number: (916-315-5081)

3-yr Cost of Ownership: **13,912,046**
QphH@10000GB: **86,282.7**
$/QphH@10000GB: **$ 161**

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

Audited By: Francois Raab for InfoSizing (www.sizing.com)

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

HP Integrity Superdome Enterprise Server

TPC-H Rev 2.1.0

Report Date: December 24, 2004

Measurement Results

Database Scaling (SF/size)	10000
Total Data Storage/Database Size	10.53
Start of Database Load Time	2004-09-17 01:11:42
End of Database Load Time	2004-09-17 08:30:08
Database Load Time	07:19:00
Query Streams for Throughput Test (S)	9
TPC-H Power	112,622.1
TPC-H Throughput	66,103.4
TPC-H Composite Query-per-Hour Metric (QphH@10000GB)	86,282.7
Total System Price Over 3 Years	$13,912,047
TPC-H Price/Performance Metric ($/QphH@10000GB)	$161

Measurement Intervals

Measurement Interval in Throughput Test (Ts)	107,831

Duration of Stream Execution:

	SEED	Start Date/Time	End Date/Time	Duration
Stream 00	917083008	9/17/04 09:01:44	9/17/04 12:34:40	3:32:56
Stream 01	917083009	9/17/04 12:34:40	9/18/04 13:49:26	25:14:46
Stream 02	917083010	9/17/04 12:34:40	9/18/04 17:20:51	28:46:11
Stream 03	917083011	9/17/04 12:34:40	9/18/04 17:05:42	28:31:02
Stream 04	917083012	9/17/04 12:34:40	9/18/04 15:11:00	26:36:20
Stream 05	917083013	9/17/04 12:34:40	9/18/04 17:05:09	28:30:29
Stream 06	917083014	9/17/04 12:34:40	9/18/04 16:51:40	28:17:00
Stream 07	917083015	9/17/04 12:34:40	9/18/04 15:06:36	26:31:56
Stream 08	917083016	9/17/04 12:34:40	9/18/04 16:27:16	27:52:36
Stream 09	917083017	9/17/04 12:34:40	9/18/04 11:00:50	22:26:10
Refresh		9/18/04 17:20:51	9/18/04 18:31:52	1:11:01

HP Integrity Superdome Enterprise Server

TPC-H Rev 2.1.0

Report Date December 24, 2004

TPC-H Timing Intervals (in seconds)

Duration of stream execution:

Stream ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8a	Q9	Q10	Q11	Q12
Stream 00	946.4	125.4	103.6	83.2	395.0	77.8	478.5	293.0	1300.6	166.8	288.3	243.8
Stream 01	5304.6	630.6	511.9	587.8	3486.4	218.1	3810.7	2284.6	9632.5	1408.7	1280.2	1332.0
Stream 02	4582.2	503.6	1149.9	1044.0	3461.5	150.9	3060.6	1291.1	22708.8	1925.7	1767.6	1164.1
Stream 03	4965.2	551.5	146.3	1058.4	5013.7	599.9	4772.6	724.8	20232.6	2121.8	1479.4	408.6
Stream 04	4157.9	484.5	507.4	914.3	1987.7	535.9	3022.5	1764.6	20590.8	1006.4	1368.7	1977.6
Stream 05	5014.3	933.8	1230.0	585.6	4307.8	495.3	5815.6	1910.2	12171.9	1611.7	1707.3	2284.7
Stream 06	5879.0	732.3	1248.7	524.3	3856.6	375.2	3649.2	1546.7	20026.3	886.1	1248.9	2876.3
Stream 07	4125.6	519.3	1116.1	690.9	3920.3	410.5	5572.0	1956.8	18808.9	1909.0	1203.6	1715.4
Stream 08	6595.6	710.6	892.4	634.4	4200.4	328.4	5893.6	1667.5	24465.9	2058.0	765.7	1946.3
Stream 09	3893.9	730.2	1014.6	433.9	3428.3	344.7	5156.8	3395.3	8089.2	1305.4	1179.3	2392.7
Minimum	3893.9	484.5	146.3	433.9	1987.7	150.9	3022.5	724.8	8089.2	886.1	765.7	408.6
Average	4946.5	644.0	868.6	719.3	3740.3	384.3	4528.2	1837.9	17414.1	1581.4	1333.4	1788.6
Maximum	6595.6	933.8	1248.7	1058.4	5013.7	599.9	5893.6	3395.3	24465.9	2121.8	1767.6	2876.3

Stream ID	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	RF1	RF2
Stream 00	1634.2	84.6	149.8	234.7	638.0	2430.9	492.2	243.3	1657.8	220.0	222.4	263.3
Stream 01	9091.5	762.4	2730.5	2834.8	2986.8	19971.1	3587.4	1258.9	15039.4	2133.7	215.7	250.3
Stream 02	7238.7	579.2	2030.1	1933.1	4263.5	26253.7	4327.4	1746.5	10972.0	1416.8	212.0	258.6
Stream 03	7734.3	376.3	2423.2	547.2	2259.4	29172.4	1915.4	1611.6	12942.8	1603.9	221.0	255.4
Stream 04	7024.5	479.8	2117.3	1654.5	1885.7	23796.9	2424.6	1531.6	15374.3	1172.0	217.4	261.5
Stream 05	10301.0	375.3	2595.5	2934.6	2090.5	26459.1	2407.2	1637.8	13435.4	2324.4	220.1	253.7
Stream 06	6749.1	427.3	4590.5	1892.9	2889.7	23288.9	2335.0	2031.1	13383.0	1382.5	218.4	256.8
Stream 07	6731.8	737.0	3516.9	2182.9	1788.5	20956.2	3079.8	1675.0	11488.2	1410.8	221.6	254.1
Stream 08	8839.1	264.0	4430.0	3096.7	2140.6	11491.6	3096.9	1300.3	13190.0	2347.4	217.5	252.9
Stream 09	7065.8	469.4	2654.3	2332.9	2310.3	17646.1	2868.2	1737.2	10416.7	1904.3	217.6	254.7
Minimum	6731.8	264.0	2030.1	547.2	1788.5	11491.6	1915.4	1258.9	10416.7	1172.0	212.0	250.3
Average	7864.0	496.7	3009.8	2156.6	2512.8	22115.1	2893.5	1614.4	12915.8	1744.0	217.9	255.3
Maximum	10301.0	762.4	4590.5	3096.7	4263.5	29172.4	4327.4	2031.1	15374.3	2347.4	221.6	261.5

October 6, 2004

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

Platform: **HP Integrity Superdome Enterprise Server**

Database Manager: Oracle Database 10g Enterprise Edition w/ Partitioning

Operating System: HP-UX 11.i V2 64-bit

The results were:

CPU (Speed)	Memory	Disks	QphH@10,000GB
HP Integrity Superdome Enterprise Server			
128 x Intanium2 (1.5 GHz)	**6 MB L3-Cache/cpu 512 GB Main**	**1452 x 73 GB 8 x 36 GB**	**86,282.7**

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:

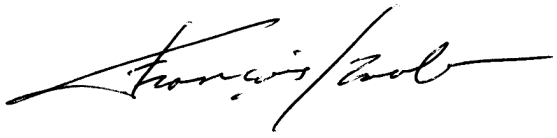
- The database records were defined with the proper layout and size
- The database population was generated using DBGEN
- The database was properly scaled to 10 TB and populated accordingly
- The compliance of the database auxiliary data structures was verified
- The database load time was correctly measured and reported
- The required ACID properties were verified and met
- The query input variables were generated by QGEN

- The query text was produced using minor modifications and the approved variant 8a
- The execution of the queries against the SF1 database produced compliant answers
- A compliant implementation specific layer was used to drive the tests
- The throughput tests involved 9 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.
- At least 8 hours of database log was configured
- The system pricing was verified for major components and maintenance
- The major pages from the FDR were verified for accuracy

Additional Audit Notes:

None.

Respectfully Yours,



François Raab
President

1 General Items

1.1 Benchmark Sponsor

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

Hewlett-Packard Company is the test sponsor of this TPC Benchmark H benchmark.

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:

Database Tuning Options

Optimizer/Query execution options

Query processing tool/language configuration parameters

Recovery/commit options

Consistency/locking options

Operating system and configuration parameters

Configuration parameters and options for any other software component incorporated into the pricing structure;

Compiler optimization options.

Appendix A contains the HP-UX and Oracle Database 10g Enterprise Edition with Partitioning parameters used in this benchmark.

1.3 Configuration Diagrams

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

Measured Configuration:

- 128 1.5GHz Itanium2 CPUs each with 6MB L3 Cache.
- 512 GB Memory
- 176 PCI Fibre Channel 2X Cards
- 4 I/O Expansion Cabinets
- 2 HP 1000 BaseSX PCI Lan Adapters
- 22 HP Disk Array XP128 (with a total of 1452 73GB Disks)
- 2 High Availability Storage Systems (with a total of 8 9 GB 1-36GB LP 10K LVD SE U320 HDD Disks)
- 2 DVD ROM
- 2 SCSI Cards
- 8 HP PCI-X 2-port 4x Fabric (HCA Adapter)
- 1 24-port 4x Fabric Copper Switch

Priced Configuration:

- 128 1.5GHz Itanium2 CPUs each with 6MB L3 Cache.
- 512 GB Memory
- 176 PCI Fibre Channel 2X Cards

- 4 I/O Expansion Cabinets
- 2 HP 1000 BaseSX PCI Lan Adapters
- 22 HP Disk Array XP128 (with a total of 1452 73GB Disks)
- 2 HP Surestore Disk System 2100 (with a total of 8 36 GB 1-36GB LP 10K LVD SE U320 HDD Disks)
- 2 DVD ROM
- 2 SCSI Cards
- 8 HP PCI-X 2-port 4x Fabric (HCA Adapters)
- 1 24-port 4x Fabric Copper Switch

The difference between measured and priced is a High Availability Storage System for the root disk which currently is obsolete. For the priced system we used a Surestore Disk System 2100 was substituted.

Terminal

Keyboard

Mouse

2 HP Integrity Superdome Enterprise Servers

HP 12-port 4X Fabric Copper Switch

EACH WITH:

64 – 1.5GHz Itanium2 Processors
256GB Memory
88 PCI Fibre Channel 2x Cards
2 I/O Expansion Cabinets
1 HP 1000 BaseSX PCI Lan Adapter
1 DVD ROM
1 SCSI Card
4 HP PCI-X 2-port 4X Fabric (HCA Adapters)
1 HP Surestore Disk System 2100 with 4 36GB Disks

Priced Configuration

22 StorageWorks XP128 Disk Arrays

- with 1452 XP128 73GB 10k RPM Disks

Terminal Keyboard Mouse

2 HP Integrity Superdome Enterprise Servers

HP 12-port 4X Fabric Copper Switch

EACH WITH:

- **64 – 1.5GHz Itanium2 Processors**
- **256GB Memory**
- **88 PCI Fibre Channel 2x Cards**
- **2 I/O Expansion Cabinets**
- **1 HP 1000 BaseSX PCI Lan Adapter**
- **1 DVD ROM**
- **1 SCSI Card**
- **4 HP PCI-X 2-port 4X Fabric (HCA Adapters)**
- **1 High Availability Storage System with 4 9.1GB Disks**

Measured Configuration

22 StorageWorks XP128 Disk Arrays

- with 1452 XP128 73GB 10k RPM Disks

2 Clause 1 Logical Database Design Related Items

2.1 Database Definition Statements

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

Appendix B describes the scripts that define, create, and analyze the tables and indices for the TPC-H database.

2.2 Physical Organization

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

No record clustering or index clustering was used. Columns were reordered in the tables – please refer to the table create statements for the ordering.

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 base and index tables except NATION and REGION. The details of this partitioning can be understood by examining the syntax of the table and index definition statements in Appendix B. Similar partitioning was used in the qualification database size.

Section 5.2 describes the distribution of tables and logs across all media.

2.4 Replication

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

No replication was used.

3 Clause 2 Queries and Refresh Functions

3.1 Query Language

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

SQL was the query language used to implement all queries.

3.2 Verifying Method for Random Number Generation

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

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

3.3 Generating Values for Substitution Parameters

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

QGEN version 1.3.0 was used to generate the substitution parameters.

3.4 Query Text and Output Data from Qualification Database

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

- Appendix C contains the actual query text and query output.

3.5 Query Substitution Parameters and Seeds Used

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

Appendix E contains the seed and query substitution parameters.

3.6 Query Isolation Level

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

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

3.7 Source Code of Refresh Functions

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

The refresh function is part of the implementation-specific layer/driver code included in Appendix F.

4 Clause 3 Database System Properties

4.1 ACID Properties

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

Source code for ACID test is included in Appendix C.

4.2 Atomicity

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

Completed Transaction

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

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

Aborted Transaction

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

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

4.3 Consistency

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

Consistency Test

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

1. The consistency of the ORDERS and LINEITEM tables was verified based on a sample of order keys.
2. 100 ACID Transactions were submitted from each of 9 execution streams.
3. The consistency of the ORDERS and LINEITEM tables was re-verified.

4.4 Isolation

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

Read-Write Conflict with Commit

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

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

Read-Write Conflict with Rollback

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

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

Write-Write Conflict with Commit

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

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

Write-Write Conflict with Rollback

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

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

Concurrent Progress of Read and Write on Different Tables

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

1. An ACID Transaction, T1, was started for a randomly selected O_KEY, L_KEY, and DELTA. T1 was suspended prior to COMMIT.

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

Read-Only Query Conflict with Update Transactions

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

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

4.5 Durability

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

Failure of a Durable Medium

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

The disks containing TPC-H tables and log files were on RAID 1/0 protected disk groups. During the durability test, one disk was removed from each RAID group containing the data and the log. One of the cache boards on the array was also failed. The test continued uninterrupted, because of the RAID and cache mirroring protection on the array.

System Crash

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

The system crash and memory failure tests were combined. Power to both the servers was turned off during the durability test. When power was restored, the system rebooted and the database was restarted. The durability success file and the HISTORY table were compared and the counts matched.

Memory Failure

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

See the previous section.

5 Clause 4 Scaling and Database Population

5.1 Ending Cardinality of Tables

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

Table	Cardinality
ORDER	15,000,000,000
LINEITEM	59,999,994,267
CUSTOMER	1,500,000,000
PART	2,000,000,000
SUPPLIER	100,000,000
PARTSUPP	8,000,000,000
NATION	25
REGION	5

5.2 Distribution of Tables and Logs Across Media

Distribution of tables and logs across media:

The 64 disks on each XP128 were configured into 16 “array groups” (4 disks per each RAID 1/0 group). Array group was divided into 7 luns:

LUN1 for lineitem and orders table

LUN2 for remaining tables and indexes

LUN3 for temp space

LUN4 for log, sys, undo

LUN5 for flat files data

LUN6 for swap

LUN7 unused, except for ACID tests

32 logical volumes, each spanning 11 LUN1s were created and each logical volume was divided into a total of 42 lvols. The lineitem tables was loaded across 21 lvols from each of the 32 logical volumes. The orders table was similary loaded across the remaining 21 lvols from each of the 32 logical volumes.

32 logical volumes, each spanning 11 LUN2s were created and divided into a number of lvols for holding the remaining tables and indexes.

32 logical volumes, each spanning 11 LUN3s were created and the temp space was evenly configured across all the volumes.

8 logical volumes were created across 44 LUN4s each and these volume groups were used to hold log, undo and sys.

Swap space for the first node was allocated from a volume group across 176 LUN6s and the swap space for the second node was allocated from another volume group across the remaining 176 LUN6s.

OS root and the Oracle home directory were configured on two disks from the JBOD array on both the nodes.

5.3 Database Partition/Replication Mapping

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

Horizontal partitioning was used for all base and index tables except NATION and REGION. The details of this partitioning can be understood by examining the syntax of the table and index definition statements in Appendix B. Similar partitioning was used in the qualification database size.

Section 5.2 describes the distribution of tables and logs across all media..

5.4 RAID Feature

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

RAID1/0 was used for log, data, temp, index, and all other.

5.5 DBGEN Modification

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

The supplied DBGEN version 1.3.0 was not modified to generate the database population for this benchmark.

5.6 Database Load Time

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

The database load time was 07:19:00.

5.7 Data Storage Ratio

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

The data storage ratio is computed from the following information:

Type	Quantity	Disk Size	Total
2 HP Surestore Disk System 2100	8	36	288
22 HP Disk Array XP128	1452	73	105,996.0
TOTAL			**106,284.0**
Scale Factor			**10,000**
Storage Ratio			**10.63**

5.8 Database Load Mechanism Details and Illustration

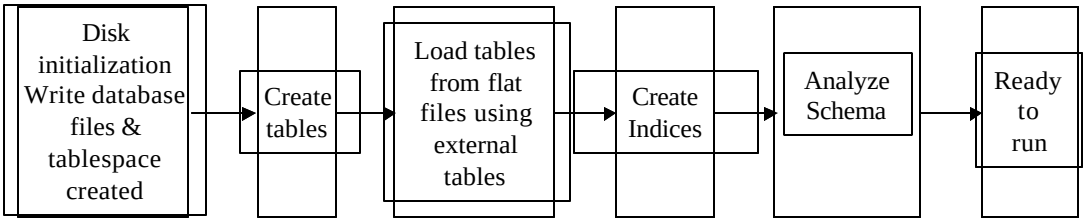
The details of the database load must be described, including a block diagram illustrating the overall process.

The database was loaded using data generation stored on the flat files all on the tested and priced configuration

5.9 Qualification Database Configuration

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

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



6 Clause 5 Performance Metrics and Execution-Rules

6.1 System Activity Between Load and Performance Tests

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

A script was run to display the hardware configurations of the SUT.

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

The database was restarted.

All scripts and queries used are included in Appendix E.

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. Database started
2. RF1 Refresh Transaction
3. Stream 00 Execution
4. RF2 Refresh Transaction

6.3 Timing Intervals for Each Query and Refresh Functions

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

The timing intervals for each query and both update functions are given in the Numerical Quantities Summary earlier in this document.

6.4 Number of Streams for the Throughput Test

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

9 streams were used for the throughput test.

6.5 Start and End Date/Time of Each Query Stream

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

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

6.6 Total Elapsed Time of the Measurement Interval

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

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

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

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

		RF1		RF2	
Number	**Date**	**Start**	**End**	**Start**	**End**
1	9/18/04	17:20:51	17:24:27	17:24:27	17:28:37
2	9/18/04	17:28:37	17:32:09	17:32:09	17:36:28
3	9/18/04	17:36:28	17:40:09	17:40:09	17:44:25
4	9/18/04	17:44:25	17:48:02	17:48:02	17:52:24
5	9/18/04	17:52:24	17:56:04	17:56:04	18:00:18
6	9/18/04	18:00:18	18:03:56	18:03:56	18:08:13
7	9/18/04	18:08:13	18:11:55	18:11:55	18:16:09
8	9/18/04	18:16:09	18:19:46	18:19:46	18:23:59
9	9/18/04	18:23:59	18:27:37	18:27:37	18:31:52

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

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

The timing intervals for each query and each update function are given in the Numerical Quantities Summary earlier in this document.

6.9 Performance Metrics

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

The performance metrics, and the numbers, on which they are based, is given in the Numerical Quantities Summary earlier in this document.

6.10 The Performance Metric and Numerical Quantities from Both Runs

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

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

	QppH@10000GB	**QthH@10000GB**	**QphH@10000GB**
Reported Run	112,622.1	66,103.4	86,282.7
Reproducibility Run	114,439.3	66,729.1	87,386.7
% Difference	1.6%	0.9%	1.3%

6.11 System Activity Between Performance Tests

Any activity on the SUT that takes place between the conclusion of the Reported Run and the beginning of Reproducibility Run must be disclosed.

The database was restarted between the two runs.

7 Clause 6 SUT and Driver Implementation Related Items

7.1 Driver

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

All stream executions are performed by a single script. QGEN is used to produce query text.

For each power-test run:

- The SQL for RF1 is submitted to the database
- Then the queries as generated by QGEN are submitted in the order defined by Clause 5.3.5.4
- The SQL for RF2 is submitted to the database**.**

7.2 Implementation-Specific Layer (ISL)

If an implementation specific layer is used, then a detailed description of how it performs its functions must be provided. All related source code, scripts and configuration files must be disclosed. The information provided should be sufficient for an independent reconstruction of the implementation specific layer.

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

7.3 Profile-Directed Optimization

If profile-directed optimization as described in Clause 5.2. is used, such use must be disclosed..

Profile-directed optimization subject to the requirements of *5.2.9* and *5.2.10* was not used.

8 Clause 7 Pricing

8.1 Hardware and Software Used in the Priced System

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

A detailed list of hardware and software used in the priced system is included in the pricing sheet in the executive summary. All prices are currently effective.

8.2 Total Three Year Price

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

A detailed pricing sheet of all the hardware and software used in this configuration and the 3-year maintenance costs, demonstrating the computation of the total 3-year price of the configuration, is included in the executive summary at the beginning of this document.

8.3 Availability Date

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

Availability Dates:

Server Hardware	Now
Server Software	Now
Storage	Now
Database Manager (Oracle Database 10g Enterprise Edition with Partitioning)	April 6, 2005

9 Clause 8 Auditor's Information and Attestation Letter

9.1 Auditor's Report

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

This implementation of the TPC Benchmark H was audited by Francois Raab for InfoSizing Further information regarding the audit process may be obtained from:

Francois Raab
InfoSizing
1373 N. Franklin Steet
Colorado Springs, CO 80903
(719) 473-7555
(719) 473-7554

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

10 Report Availability

Requests for this TPC Benchmark H Full Disclosure Report should be sent to:

Appendix A Parameter Settings

A. 1 10TB-rac1.ora

```
cluster_database = true
cluster_database_instances = 2
instance_number = 1
instance_name = tpch1
thread = 1
undo_tablespace = ts_undo1
ifile = /oracle/dbs/10TB_titan.ora
```

A. 2 10TB-rac2.ora

```
cluster_database = true
cluster_database_instances = 2
instance_number = 2
instance_name = tpch2
thread = 2
undo_tablespace = ts_undo2
ifile = /oracle/dbs/10TB_titan.ora
```

A. 3 10TB-titan.ora

```
aq_tm_processes = 0
audit_trail = FALSE
compatible = 10.1.0.2
control_files =
(/dbms/links/control1,/dbms/links/control2)
db_block_checksum = false
db_block_size = 8192
db_cache_size = 12g
db_file_multiblock_read_count = 256
db_files = 2400
db_name = 10tb
db_writer_processes = 16
dml_locks = 40000
enqueue_resources = 40000
global_names = FALSE
hpux_sched_noage = 180
log_buffer = 33554432
log_checkpoints_to_alert = true
max_dump_file_size = unlimited
nls_date_format = YYYY-MM-DD
open_cursors = 1024
optimizer_dynamic_sampling = 3
optimizer_features_enable = 10.1.0.1
optimizer_index_cost_adj = 200
optimizer_mode = CHOOSE
parallel_adaptive_multi_user = true
parallel_execution_message_size = 16384
parallel_max_servers = 1024
parallel_min_servers = 850
pga_aggregate_target = 75g
processes = 5000
replication_dependency_tracking = false
shared_pool_size = 9g
sort_area_size = 1073741824
statistics_level = BASIC
undo_management = auto
undo_retention = 200000
```

A. 4 system

```
*
* Created on Wed Sep 15 14:08:17 2004
*
version 1
configuration nextboot "booted from 'kernel_big_nfile'" [4148af40]
*
* Module entries
*
module pppoe loaded 1.0.0
module mip6mod best [4043C689]
module ipoib best [40DC7F2C]
module ipf loaded 0.1.0
module mpt best [3F4A8371]
module vols best [3F41B706]
module vol best [3F41B706]
module vxdmp best [3F41B577]
module vxvm best [3F41B706]
module lv best [3F559170]
module lvm best [3F559170]
module vxportal best [412D5F4A]
module vxfs best [412D5F4A]
module pfil auto 0.1.0
module igelan best [3F454271]
module iether best [3F4542A0]
module gelan best [3F454178]
module fddi4 best [3F4122D1]
module td best [3F533FD9]
module cifs best [3F465E27]
module pckt best [3F559170]
module ptm best [3F559170]
module pts best [3F559170]
module ptem best [3F559170]
module ldterm best [3F559170]
module ffs best [3F559170]
module pipemod best [3F559170]
module pipedev best [3F559170]
module tirdwr best [3F559170]
module timod best [3F559170]
module sc best [3F559170]
module echo best [3F559170]
module sad best [3F559170]
module strlog best [3F559170]
module clone best [3F559170]
module hpstreams best [3F559170]
module cachefsc best [3F559170]
module autofsc best [3F559170]
module rpcmod best [3F559170]
module nfsm best [3F559170]
module nfs_client best [3F559170]
module nfs_server best [3F559170]
module nfs_core best [3F559170]
module nms best [4032B8DD]
module netdiag1 best [3F56E2F0]
module token_arp best [40369D5B]
module dlpi best [40369D5B]
module intl100 best [3F559170]
module btlan best [3F559170]
```

```
module tels best [3F559170]
module telm best [3F559170]
module tun best [4032B8EB]
module uipc best [4043C647]
module inet best [404E7FCF]
module rng loaded 0.1.0
module cdfs best 0.1.0
module dev_config best [3F56E2F0]
module dmem best [3F56E2F0]
module diag2 best [3F56E2F0]
module c8xx best [3F56E2F0]
module pdh best [3F56E2F0]
module lion_psm best [3F56E2F0]
module ia64_psm best [3F56E2F0]
module wxb_hp best [3F56E2F0]
module sac best [3F56E2F0]
module acpi_node best [3F56E2F0]
module LCentIf best [3F56E2F0]
module ipmi best [3F56E2F0]
module pty1 best [3F56E2F0]
module pty0 best [3F56E2F0]
module azusa_psm best [3F56E2F0]
module PCItoPCI best [3F56E2F0]
module stape best [3F56E2F0]
module fcpdev best [40A94028]
module fcparray best [40A94028]
module fcp best [40A94028]
module sctl best [3F56E2F0]
module sdisk best [3F56E2F0]
module tgt best [3F56E2F0]
module asio0 best [3F56E2F0]
module lba best [3F56E2F0]
module sba best [3F56E2F0]
module cell best [3F56E2F0]
module root best [3F56E2F0]
module ib_subsys best [4134F907]
module ib_t best [412BF0BF]
*
* Swap entries
*
*
* Dump entries
*
dump lvol
*
* Driver binding entries
*
*
* Tunables entries
*
tunable bufpages 1000000
tunable STRMSGSZ 65535
tunable create_fastlinks 1
tunable dbc_max_pct 3
tunable dbc_min_pct 3
tunable eqmemsize 512
tunable hfs_max_ra_blocks 20
tunable hfs_max_revra_blocks 20
tunable hfs_ra_per_disk 256
tunable hfs_revra_per_disk 256
tunable max_async_ports 2048
tunable max_thread_proc 2048
tunable maxdsiz 0x40000000
tunable maxdsiz_64bit 0x80000000
tunable maxfiles 4096
tunable maxfiles_lim 4096
tunable maxssiz 0x10000000
tunable maxssiz_64bit 268435456
tunable maxtsiz 1073741824
tunable maxtsiz_64bit 4294967296
tunable maxuprc 3277
tunable maxvgs 200
tunable msgmap 5122
tunable msgmax 32768
tunable msgmnb 65536
tunable msgmni 512
tunable msgseg 20480
tunable msgssz 128
tunable msgtql 5120
tunable nfile 2000000
tunable ninode 120000
tunable nproc 4096
tunable npty 200
tunable nstrpty 200
tunable nswapdev 25
tunable semmni 4096
tunable semmns 8192
tunable semmnu 4092
tunable semume 512
tunable semvmx 32768
tunable shmmax 0x40000000000
tunable shmmni 2048
tunable shmseg 512
tunable swapmem_on 0
tunable swchunk 65536
tunable timezone 480
tunable unlockable_mem 1
tunable vps_ceiling 64
tunable vxfs_ifree_timelag 3600000
```

A. 5 env

```
########## MACHINE PARAMETERS
####################
export RAC_NODES="titan1 titan2"
########## PATHS ###################################
export KIT_DIR=/dbms/oracle10i/kit
export SCHEMA_DIR=$KIT_DIR/schema
export PERL=/opt/perl/bin/perl
#export BUMPX_DIR=$KIT_DIR/bumpx
#export BUMPX_OUT=$KIT_DIR/bumpx
export UTILS=$KIT_DIR/utils
export TEST_DB=/tmp
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=/dbms/oracle10i/kit/acid/answers2validate
export ACID_OUT=$KIT_DIR/out
export DSS_CONFIG=$DBGEN
export DSS_QUERY=$KIT_DIR/queries
export DSS_PATH=$ADE_VIEW_ROOT
export MAINT=$KIT_DIR/maintenance
export CC=/opt/ansic/bin/cc
export FRAME=$KIT_DIR/frame
export FRAME_DIR=/dbms/oracle10i/frame
#export REGR_TEST=$KIT_DIR/internal/regression_test
export SCALE_FACTOR=10000
export UPDATE_1_DOP=64
export UPDATE_2_DOP=256
############ FRAME STUFF
export FRAME_PATH=$KIT_DIR/frame

#export ORACORE3INCL=/vobs/oracore3/include
#export ORACORE3PUBL=/vobs/oracore3/public
```

```
export ORACORE3INCL=$ORACLE_HOME/rdbms/demo
export ORACORE3PUBL=$ORACLE_HOME/rdbms/public
#export RDBMSPUBL=/vobs/rdbms/public
export RDBMSPUBL=$ORACLE_HOME/rdbms/public
#export NETWORKPUBL=/vobs/network_src/public
export NETWORKPUBL=$ORACLE_HOME/network/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}:${
PATH}
#
########## ENVIRONMENT VARIABLES
#################
export WORKLOAD=TPCH
export HOST=
#export OPTLEVEL=X02
export GETOPT=-DSTDLIB_HAS_GETOPT
export PLATFORM=
#export INITORA=$KIT_DIR/schema/test_db/testdb.ora
#export INITORA=$KIT_DIR/schema/test_db/sf100.ora

########## 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
DATABASE_USER=tpch/tpch
```

A. 6 profile (1st node)

```
stty erase "^H" kill "^x" intr "^C" eof "^D" susp "^z"
export EDITOR=/usr/bin/vi
export ORACLE_HOME=/oracle

export ORACLE_SID=tpch1
echo 'ORACLE_SID is tpch1'
export KIT_DIR=/dbms/oracle10i/kit

#export ORACLE_SID=1gtpch1
#echo 'ORACLE_SID is 1gtpch1'
#export KIT_DIR=/dbms/oracle10i/kit_acid

export
SHLIB_PATH=/oracle/lib:/oracle/lib32::/oracle/rdbms/lib:/oracle/ne
twork/lib
export
LD_LIBRARY_PATH=/oracle/lib:/oracle/lib64:/oracle/rdbms/lib:/o
racle/network/lib64
export SAVEHIST=2049
export FRAME_PATH=/dbms/oracle10i/frame
export O=/dbms/oracle10i
export ORACLE_PATH=/dbms/oracle10i/frame/tools
export PS1="`whoami`-(`hostname`)> "
export skgxp_trace_path=/tmp/srq.tpch1
export
PATH=./:/oracle/bin:/oracle:/oracle/lib:/tools:/tpch/run_power:/tpch:
/dbms/oracle10i/frame/bin:/dbms/oracle10i/frame:/dbms/oracle10i/to
ols/bin:/tools/Tusc:/dbms/tpcd_v8/bumpx/bumpx:/dbms/tpcd_v8/bu
mpx/dbgen:/dbms/tpcd_v8/out/scripts:/opt/ansic/bin:/opt/langtools/b
in:/sbin:/usr/sbin:.:/bin:/usr/bin:/usr/local/bin:/usr/contrib/bin:/et c:/us
r/include:/dbms/oracle10i/kit:/dbms/oracle10i/kit/bumpx:/dbms/oracl
e10i/local/TestIO:/usr/ccs/bin/:/opt/caliper/bin:/opt/rdma/bin:~/bin

alias ltt="ls -ltr |tail -30"
alias cd_frame="cd /dbms/oracle10i/frame"
alias cd_stats="cd /dbms/oracle10i/frame/stats"
alias cd_q="cd /dbms/oracle10i/frame/queries/queries_tpch"
alias cd_log="cd /oracle/rdbms/log"
alias cd_u="cd /dbms/oracle10i/frame/queries/queries_tpch/updates"
alias ltm="ls -lt |more"
alias cdbin="cd /dbms/tpcd_v8/bin"
alias cdlog="cd /oracle/rdbms/log"
alias cdtools="cd /dbms/oracle10i/tools/bin"
alias cdq="cd /tpch/tpch/run_power"
alias pso="ps -ef | grep ora | grep -v sleep"
alias pso_hc="ps -fu oracle | sort -n -k2"
alias setterm="TERM=dtterm;export TERM"
alias taillog="tail -f /oracle/rdbms/log/alert_$ORACLE_SID.log"
alias ftp_oracle="/usr/bin/ftp 148.87.8.191"
alias cdlog="cd $ORACLE_HOME/rdbms/log"

umask 002
iosum(){
if [ "$1" -eq "" ] ; then
   echo usage:  iosum iterations
else
 sar -d 5 $1 | ${FRAME_PATH}/bin/io.pl
fi
}
```

A. 7 profile (2nd node)

```
stty erase "^H" kill "^x" intr "^C" eof "^D" susp "^z"

export EDITOR=/usr/bin/vi
export ORACLE_HOME=/oracle

export ORACLE_SID=tpch2
echo 'ORACLE_SID is tpch2'
export KIT_DIR=/dbms/oracle10i/kit

#export ORACLE_SID=1gtpch2
#echo 'ORACLE_SID is 1gtpch2'
#export KIT_DIR=/dbms/oracle10i/kit_acid

export
SHLIB_PATH=/oracle/lib:/oracle/lib32::/oracle/rdbms/lib:/oracle/ne
twork/lib
export
LD_LIBRARY_PAT H=/oracle/lib:/oracle/lib64:/oracle/rdbms/lib:/o
racle/network/lib64
export SAVEHIST=2049
export FRAME_PATH=/dbms/oracle10i/frame
export O=/dbms/oracle10i
export ORACLE_PATH=/dbms/oracle10i/frame/tools
export PS1="`whoami`-(`hostname`)> "
export  skgxp_trace_path=/tmp/srq.tpch2
export
PATH=./:/oracle/bin:/oracle:/oracle/lib:/tools:/tpch/run_power:/tpch:
/dbms/oracle10i/frame/bin:/dbms/oracle10i/frame:/dbms/oracle10i/to
ols/bin:/tools/Tusc:/dbms/tpcd_v8/bumpx/bumpx:/dbms/tpcd_v8/bu
```

```
mpx/dbgen:/dbms/tpcd_v8/out/scripts:/opt/ansic/bin:/opt/langtools/b
in:/sbin:/usr/sbin:.:/bin:/usr/bin:/usr/local/bin:/usr/contrib/bin:/etc:/us
r/include:/dbms/oracle10i/kit:/dbms/oracle10i/kit/bumpx:/dbms/oracl
e10i/local/TestIO:/usr/ccs/bin/:/opt/caliper/bin:/opt/rdma/bin:~/bin

alias ltt="ls -ltr |tail -30"
alias cd_frame="cd /dbms/oracle10i/frame"
alias cd_stats="cd /dbms/oracle10i/frame/stats"
alias cd_q="cd /dbms/oracle10i/frame/queries/queries_tpch"
alias cd_log="cd /oracle/rdbms/log"
alias cd_u="cd /dbms/oracle10i/frame/queries/queries_tpch/updates"
alias ltm="ls -lt |more"
alias cdbin="cd /dbms/tpcd_v8/bin"
alias cdlog="cd /oracle/rdbms/log"
alias cdtools="cd /dbms/oracle10i/tools/bin"
alias cdq="cd /tpch/tpch/run_power"
alias pso="ps -ef | grep ora | grep -v sleep"
alias pso_hc="ps -fu oracle | sort -n -k2"
alias setterm="TERM=dtterm;export TERM"
alias taillog="tail -f /oracle/rdbms/log/alert_$ORACLE_SID.log"
alias ftp_oracle="/usr/bin/ftp 148.87.8.191"
alias cdlog="cd $ORACLE_HOME/rdbms/log"

umask 002
iosum(){
if [ "$1" -eq "" ] ; then
    echo usage:  iosum iterations
else
 sar -d 5 $1 | ${FRAME_PATH}/bin/io.pl
fi
}
```

Appendix B Build Programs and Scripts

B.1 dbcre.sh

```
#!/bin/ksh

echo START CREATE DB at `date`

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

shutdown abort;


startup pfile=/oracle/dbs/10TB_init.ora nomount;
create database
  controlfile reuse
  logfile '/dbms/links/log_1' size 32000m reuse,
          '/dbms/links/log_2' size 32000m reuse
  datafile '/dbms/links/sys_1' size 2000m reuse
  sysaux datafile '/dbms/links/aux' size 2000m reuse
  undo tablespace ts_undo1
       datafile  '/dbms/links/undo_1' size 32000m reuse
  default temporary tablespace ts_temp
       tempfile '/dbms/links/temp_1' size 32000m reuse
       extent management local uniform size 5m
  maxdatafiles 1000
  maxinstances 2
;

create undo tablespace ts_undo2
       datafile  '/dbms/links/undo_21' size 32000m reuse
;

alter database add logfile thread 2
  '/dbms/links/log_3' size 32000m reuse,
  '/dbms/links/log_4' size 32000m reuse;
alter database enable public thread 2;

set termout off
set echo off
spool /tmp/cat
@?/rdbms/admin/catalog.sql;
@?/rdbms/admin/catproc.sql;
@?/rdbms/admin/catclust.sql;
connect system/manager
@?/sqlplus/admin/pupbld.sql;
spool off

alter system switch logfile;
!
echo END CREATE DB at `date`
```

B.2 sctso.sh

```
#!/bin/ksh


/dbms/oracle10g/frame/bin/tshut

/dbms/oracle10g/frame/bin/tstart

echo CREATE TABLESPACES at `date`

sqlplus  / as sysdba <<! &
set timing on
set echo on

--drop tablespace ts_default including contents;
create tablespace ts_default
datafile '/dbms/links/def_1' size 2800m reuse
extent management local
autoallocate
;
!


(( i = 1 ))

while (( i <= 300 ))
do
sqlplus  / as sysdba <<! &
set timing on
set echo on

--drop tablespace ts_l${i} including contents;
create tablespace ts_l${i}
datafile '/dbms/links/line_${i}' size 16500m reuse
extent management dictionary default storage (initial 1050m next 20m
maxextents unlimited pctincrease 0)
;
!

(( i = $i + 1 ))
done

wait


(( i = 301 ))

while (( i <= 672 ))
do
sqlplus  / as sysdba <<! &
set timing on
set echo on

--drop tablespace ts_l${i} including contents;
create tablespace ts_l${i}
datafile '/dbms/links/line_${i}' size 16500m reuse
extent management dictionary default storage (initial 1050m next 20m
maxextents unlimited pctincrease 0)
;
!

(( i = $i + 1 ))
done

wait


(( i = 1 ))

while (( i <= 300 ))
do
sqlplus  / as sysdba <<! &
set timing on
set echo on
```

```
--drop tablespace ts_o${i} including contents;
create tablespace ts_o${i}
datafile '/dbms/links/ord_${i}' size 4200m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
!


(( i= $i + 1 ))
done

wait


(( i = 301 ))

while (( i <= 672 ))
do
sqlplus  / as sysdba <<! &
set timing on
set echo on


--drop tablespace ts_o${i} including contents;
create tablespace ts_o${i}
datafile '/dbms/links/ord_${i}' size 4200m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
!


(( i= $i + 1 ))
done

wait


(( i = 1 ))

while (( i <= 32 ))
do
sqlplus  / as sysdba <<! &
set timing on
set echo on

--drop tablespace ts_small${i} including contents;
create tablespace ts_small${i}
datafile '/dbms/links/small_${i}' size 25000m reuse
extent management dictionary default storage (initial 450m next 20m
maxextents unlimited pctincrease 0)
;
;
!

(( i= $i + 1 ))
done


sqlplus  / as sysdba <<! &
set timing on
set echo on

--drop tablespace ts_okey including contents;
create tablespace ts_okey
datafile '/dbms/links/okey_1' size 14000m reuse
extent management local
autoallocate
;
!


sqlplus  / as sysdba <<! &
set timing on
set echo on

--drop tablespace ts_custkey including contents;
create tablespace ts_custkey
datafile '/dbms/links/custkey_1' size 1700m reuse
extent management local
uniform size 30M
;
!

sqlplus  / as sysdba <<! &
set timing on
set echo on

--drop tablespace ts_lokey including contents;
create tablespace ts_lokey
datafile '/dbms/links/lokey_1' size 16000m reuse
extent management local
autoallocate
;
!

sqlplus  / as sysdba <<! &
set timing on
set echo on


--drop tablespace ts_psupp1 including contents;
create tablespace ts_psupp1
datafile '/dbms/links/psupp_1' size 26400m reuse
extent management dictionary default storage (initial 1100m next 50m
maxextents unlimited pctincrease 0)
;
!

wait

(( i = 2 ))

while (( i <= 64 ))
do
sqlplus  / as sysdba <<! &
set timing on
set echo on

alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_${i}' size 26400m reuse;
!

(( i = $i + 1 ))
done


(( i = 2 ))

while (( i <= 32 ))
do
sqlplus  / as sysdba <<! &
set timing on
set echo on

alter tablespace ts_okey
add datafile '/dbms/links/okey_${i}' size 14000m reuse;
!

(( i = $i + 1 ))
```

```
done


(( i = 2 ))

while (( i <= 32 ))
do
sqlplus  / as sysdba <<! &
set timing on
set echo on

alter tablespace ts_custkey
add datafile '/dbms/links/custkey_${i}' size 1500m reuse;
!

(( i = $i + 1 ))
done


(( i = 2 ))

while (( i <= 128 ))
do
sqlplus  / as sysdba <<! &
set timing on
set echo on

alter tablespace ts_lokey
add datafile '/dbms/links/lokey_${i}' size 16000m reuse;
!

(( i = $i + 1 ))
done

wait

(( i = 2 ))

while (( i <= 224 ))
do
sqlplus  / as sysdba <<! &
set timing on
set echo on


alter tablespace ts_temp
add tempfile '/dbms/links/temp_${i}' size 32000m reuse;
!

(( i = $i + 1 ))
done



(( i = 2 ))

while (( i <= 20 ))
do
sqlplus  / as sysdba <<! &
set timing on
set echo on


alter tablespace ts_undo1
add datafile '/dbms/links/undo_${i}' size 32000m reuse;
!

(( i = $i + 1 ))
done


(( i = 22 ))

while (( i <= 40 ))
do
sqlplus  / as sysdba <<! &
set timing on
set echo on


alter tablespace ts_undo2
add datafile '/dbms/links/undo_${i}' size 32000m reuse;
!

(( i = $i + 1 ))
done


sqlplus  / as sysdba <<! &
set timing on
set echo on

alter tablespace ts_default
 add datafile '/dbms/links/def_2' size 2800m reuse;
!

wait

echo END CREATE TABLESPACES at `date`
```

B.3 dapop.sh

```
#!/bin/ksh

#/dbms/oracle10g/frame/bin/tshut

#/dbms/oracle10g/frame/bin/tstart

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 default tablespace ts_default;
alter user tpch temporary tablespace ts_temp;

connect tpch/tpch;
drop directory data_dir1;
drop directory data_dir2;
drop directory data_dir3;
drop directory data_dir4;
drop directory data_dir5;
drop directory data_dir6;
drop directory data_dir7;
drop directory data_dir8;
drop directory data_dir9;
drop directory data_dir10;
drop directory data_dir11;
drop directory data_dir12;
drop directory data_dir13;
```

```
drop directory data_dir14;
drop directory data_dir15;
drop directory data_dir16;
create directory data_dir1 as '/flat1/';
create directory data_dir2 as '/flat2/';
create directory data_dir3 as '/flat3/';
create directory data_dir4 as '/flat4/';
create directory data_dir5 as '/flat5/';
create directory data_dir6 as '/flat6/';
create directory data_dir7 as '/flat7/';
create directory data_dir8 as '/flat8/';
create directory data_dir9 as '/flat9/';
create directory data_dir10 as '/flat10/';
create directory data_dir11 as '/flat11/';
create directory data_dir12 as '/flat12/';
create directory data_dir13 as '/flat13/';
create directory data_dir14 as '/flat14/';
create directory data_dir15 as '/flat15/';
create directory data_dir16 as '/flat16/';


drop table l_et;
create table l_et(
   l_orderkey          number ,
   l_partkey           number ,
   l_suppkey           number ,
   l_linenumber        number ,
   l_quantity          number ,
   l_extendedprice     number ,
   l_discount          number ,
   l_tax               number ,
   l_returnflag        char(1) ,
   l_linestatus        char(1) ,
   l_shipdate          date ,
   l_commitdate        date ,
   l_receiptdate       date ,
   l_shipinstruct      char(25) ,
   l_shipmode          char(10) ,
   l_comment           varchar(44)
)
organization external (
type ORACLE_LOADER
default directory data_dir1
access parameters
(
                    records delimited by newline
          nobadfile
          nologfile
                    fields terminated by '|'
                    missing field values are null
           )
           location (
            data_dir1:'lineitem.tbl.1',
            data_dir1:'lineitem.tbl.2',
            data_dir1:'lineitem.tbl.3',
            data_dir1:'lineitem.tbl.4',
            data_dir1:'lineitem.tbl.5',
            data_dir2:'lineitem.tbl.6',
            data_dir2:'lineitem.tbl.7',
            data_dir2:'lineitem.tbl.8',
            data_dir2:'lineitem.tbl.9',
            data_dir2:'lineitem.tbl.10',
            data_dir3:'lineitem.tbl.11',
            data_dir3:'lineitem.tbl.12',
            data_dir3:'lineitem.tbl.13',
            data_dir3:'lineitem.tbl.14',
            data_dir3:'lineitem.tbl.15',
            data_dir4:'lineitem.tbl.16',
            data_dir4:'lineitem.tbl.17',
            data_dir4:'lineitem.tbl.18',
            data_dir4:'lineitem.tbl.19',
            data_dir4:'lineitem.tbl.20',
            data_dir5:'lineitem.tbl.21',
            data_dir5:'lineitem.tbl.22',
            data_dir5:'lineitem.tbl.23',
            data_dir5:'lineitem.tbl.24',
            data_dir5:'lineitem.tbl.25',
            data_dir6:'lineitem.tbl.26',
            data_dir6:'lineitem.tbl.27',
            data_dir6:'lineitem.tbl.28',
            data_dir6:'lineitem.tbl.29',
            data_dir6:'lineitem.tbl.30',
            data_dir7:'lineitem.tbl.31',
            data_dir7:'lineitem.tbl.32',
            data_dir7:'lineitem.tbl.33',
            data_dir7:'lineitem.tbl.34',
            data_dir7:'lineitem.tbl.35',
            data_dir8:'lineitem.tbl.36',
            data_dir8:'lineitem.tbl.37',
            data_dir8:'lineitem.tbl.38',
            data_dir8:'lineitem.tbl.39',
            data_dir8:'lineitem.tbl.40',
            data_dir9:'lineitem.tbl.41',
            data_dir9:'lineitem.tbl.42',
            data_dir9:'lineitem.tbl.43',
            data_dir9:'lineitem.tbl.44',
            data_dir9:'lineitem.tbl.45',
            data_dir10:'lineitem.tbl.46',
            data_dir10:'lineitem.tbl.47',
            data_dir10:'lineitem.tbl.48',
            data_dir10:'lineitem.tbl.49',
            data_dir10:'lineitem.tbl.50',
            data_dir11:'lineitem.tbl.51',
            data_dir11:'lineitem.tbl.52',
            data_dir11:'lineitem.tbl.53',
            data_dir11:'lineitem.tbl.54',
            data_dir11:'lineitem.tbl.55',
            data_dir12:'lineitem.tbl.56',
            data_dir12:'lineitem.tbl.57',
            data_dir12:'lineitem.tbl.58',
            data_dir12:'lineitem.tbl.59',
            data_dir12:'lineitem.tbl.60',
            data_dir13:'lineitem.tbl.61',
            data_dir13:'lineitem.tbl.62',
            data_dir13:'lineitem.tbl.63',
            data_dir13:'lineitem.tbl.64',
            data_dir13:'lineitem.tbl.65',
            data_dir14:'lineitem.tbl.66',
            data_dir14:'lineitem.tbl.67',
            data_dir14:'lineitem.tbl.68',
            data_dir14:'lineitem.tbl.69',
            data_dir14:'lineitem.tbl.70',
            data_dir15:'lineitem.tbl.71',
            data_dir15:'lineitem.tbl.72',
            data_dir15:'lineitem.tbl.73',
            data_dir15:'lineitem.tbl.74',
            data_dir15:'lineitem.tbl.75',
            data_dir16:'lineitem.tbl.76',
            data_dir16:'lineitem.tbl.77',
            data_dir16:'lineitem.tbl.78',
            data_dir16:'lineitem.tbl.79',
            data_dir16:'lineitem.tbl.80',
            data_dir1:'lineitem.tbl.81',
            data_dir2:'lineitem.tbl.82',
            data_dir3:'lineitem.tbl.83',
            data_dir4:'lineitem.tbl.84'
))
reject limit unlimited parallel;

drop table o_et;
create table o_et(
   o_orderkey          number ,
```

```
    o_custkey           number ,
    o_orderstatus       char(1) ,
    o_totalprice        number ,
    o_orderdate         date ,
    o_orderpriority     char(15) ,
    o_clerk             char(15) ,
    o_shippriority      number ,
    o_comment           varchar(79)
)
organization external (
type ORACLE_LOADER
default directory data_dir1
access parameters
(
                        records delimited by newline
            nobadfile
            nologfile
                        fields terminated by '|'
                        missing field values are null
             )
             location (
        data_dir1:'orders.tbl.1',
        data_dir1:'orders.tbl.2',
        data_dir1:'orders.tbl.3',
        data_dir1:'orders.tbl.4',
        data_dir1:'orders.tbl.5',
        data_dir2:'orders.tbl.6',
        data_dir2:'orders.tbl.7',
        data_dir2:'orders.tbl.8',
        data_dir2:'orders.tbl.9',
        data_dir2:'orders.tbl.10',
        data_dir3:'orders.tbl.11',
        data_dir3:'orders.tbl.12',
        data_dir3:'orders.tbl.13',
        data_dir3:'orders.tbl.14',
        data_dir3:'orders.tbl.15',
        data_dir4:'orders.tbl.16',
        data_dir4:'orders.tbl.17',
        data_dir4:'orders.tbl.18',
        data_dir4:'orders.tbl.19',
        data_dir4:'orders.tbl.20',
        data_dir5:'orders.tbl.21',
        data_dir5:'orders.tbl.22',
        data_dir5:'orders.tbl.23',
        data_dir5:'orders.tbl.24',
        data_dir5:'orders.tbl.25',
        data_dir6:'orders.tbl.26',
        data_dir6:'orders.tbl.27',
        data_dir6:'orders.tbl.28',
        data_dir6:'orders.tbl.29',
        data_dir6:'orders.tbl.30',
        data_dir7:'orders.tbl.31',
        data_dir7:'orders.tbl.32',
        data_dir7:'orders.tbl.33',
        data_dir7:'orders.tbl.34',
        data_dir7:'orders.tbl.35',
        data_dir8:'orders.t bl.36',
        data_dir8:'orders.tbl.37',
        data_dir8:'orders.tbl.38',
        data_dir8:'orders.tbl.39',
        data_dir8:'orders.tbl.40',
        data_dir9:'orders.tbl.41',
        data_dir9:'orders.tbl.42',
        data_dir9:'orders.tbl.43',
        data_dir9:'orders.tbl.44',
        data_dir9:'orders.tbl.45',
        data_dir10:'orders.tbl.46',
        data_dir10:'orders.tbl.47',
        data_dir10:'orders.tbl.48',
        data_dir10:'orders.tbl.49',
        data_dir10:'orders.tbl.50',
        data_dir11:'orders.tbl.51',
        data_dir11:'orders.tbl.52',
        data_dir11:'orders.tbl.53',
        data_dir11:'orders.tbl.54',
        data_dir11:'orders.tbl.55',
        data_dir12:'orders.tbl.56',
        data_dir12:'orders.tbl.57',
        data_dir12:'orders.tbl.58',
        data_dir12:'orders.tbl.59',
        data_dir12:'orders.tbl.60',
        data_dir13:'orders.tbl.61',
        data_dir13:'orders.tbl.62',
        data_dir13:'orders.tbl.63',
        data_dir13:'orders.tbl.64',
        data_dir13:'orders.tbl.65',
        data_dir14:'orders.tbl.66',
        data_dir14:'orders.tbl.67',
        data_dir14:'orders.tbl.68',
        data_dir14:'orders.tbl.69',
        data_dir14:'orders.tbl.70',
        data_dir15:'orders.tbl.71',
        data_dir15:'orders.tbl.72',
        data_dir15:'orders.tbl.73',
        data_dir15:'orders.tbl.74',
        data_dir15:'orders.tbl.75',
        data_dir16:'orders.tbl.76',
        data_dir16:'orders.tbl.77',
        data_dir16:'orders.tbl.78',
        data_dir16:'orders.tbl.79',
        data_dir16:'orders.tbl.80',
        data_dir5:'orders.tbl.81',
        data_dir6:'orders.tbl.82',
        data_dir7:'orders.tbl.83',
        data_dir8:'orders.tbl.84'
))
reject limit unlimited parallel;

drop table ps_et;
create table ps_et(
    ps_partkey          number ,
    ps_suppkey          number ,
    ps_availqty         number ,
    ps_supplycost       number ,
    ps_comment          varchar(199)
)
organization external (
type ORACLE_LOADER
default directory data_dir1
access parameters
(
                        records delimited by newline
            nobadfile
            nologfile
                        fields terminated by '|'
                        missing field values are null
             )
             location (
        data_dir1:'partsupp.tbl.1',
        data_dir1:'partsupp.tbl.2',
        data_dir1:'partsupp.tbl.3',
        data_dir1:'partsupp.tbl.4',
        data_dir2:'partsupp.tbl.5',
        data_dir2:'partsupp.tbl.6',
        data_dir2:'partsupp.tbl.7',
        data_dir2:'partsupp.tbl.8',
        data_dir3:'partsupp.tbl.9',
        data_dir3:'partsupp.tbl.10',
        data_dir3:'partsupp.tbl.11',
        data_dir3:'partsupp.tbl.12',
        data_dir4:'partsupp.tbl.13',
        data_dir4:'partsupp.tbl.14',
```

```
        data_dir4:'partsupp.tbl.15',
        data_dir4:'partsupp.tbl.16',
        data_dir5:'partsupp.tbl.17',
        data_dir5:'partsupp.tbl.18',
        data_dir5:'partsupp.tbl.19',
        data_dir5:'partsupp.tbl.20',
        data_dir6:'partsupp.tbl.21',
        data_dir6:'partsupp.tbl.22',
        data_dir6:'partsupp.tbl.23',
        data_dir6:'partsupp.tbl.24',
        data_dir7:'partsupp.tbl.25',
        data_dir7:'partsupp.tbl.26',
        data_dir7:'partsupp.tbl.27',
        data_dir7:'partsupp.tbl.28',
        data_dir8:'partsupp.tbl.29',
        data_dir8:'partsupp.tbl.30',
        data_dir8:'partsupp.tbl.31',
        data_dir8:'partsupp.tbl.32',
        data_dir9:'partsupp.tbl.33',
        data_dir9:'partsupp.tbl.34',
        data_dir9:'partsupp.tbl.35',
        data_dir9:'partsupp.tbl.36',
        data_dir10:'partsupp.tbl.37',
        data_dir10:'partsupp.tbl.38',
        data_dir10:'partsupp.tbl.39',
        data_dir10:'partsupp.tbl.40',
        data_dir11:'partsupp.tbl.41',
        data_dir11:'partsupp.tbl.42',
        data_dir11:'partsupp.tbl.43',
        data_dir11:'partsupp.tbl.44',
        data_dir12:'partsupp.tbl.45',
        data_dir12:'partsupp.tbl.46',
        data_dir12:'partsupp.tbl.47',
        data_dir12:'partsupp.tbl.48',
        data_dir13:'partsupp.tbl.49',
        data_dir13:'partsupp.tbl.50',
        data_dir13:'partsupp.tbl.51',
        data_dir13:'partsupp.tbl.52',
        data_dir14:'partsupp.tbl.53',
        data_dir14:'partsupp.tbl.54',
        data_dir14:'partsupp.tbl.55',
        data_dir14:'partsupp.tbl.56',
        data_dir15:'partsupp.tbl.57',
        data_dir15:'partsupp.tbl.58',
        data_dir15:'partsupp.tbl.59',
        data_dir15:'partsupp.tbl.60',
        data_dir16:'partsupp.tbl.61',
        data_dir16:'partsupp.tbl.62',
        data_dir16:'partsupp.tbl.63',
        data_dir16:'partsupp.tbl.64'
))
reject limit unlimited parallel;


drop table p_et;
create table p_et(
   p_partkey          number ,
   p_name             varchar(55) ,
   p_mfgr             char(25) ,
   p_brand            char(10) ,
   p_type             varchar(25) ,
   p_size             number ,
   p_container        char(10) ,
   p_retailprice      number ,
   p_comment          varchar(23)
)
organization external (
type ORACLE_LOADER
default directory data_dir1
access parameters
(
                        records delimited by newline
            nobadfile
            nologfile
                        fields terminated by '|'
                        missing field values are null
             )
             location (
        data_dir1:'part.tbl.1',
        data_dir2:'part.tbl.2',
        data_dir3:'part.tbl.3',
        data_dir4:'part.tbl.4',
        data_dir5:'part.tbl.5',
        data_dir6:'part.tbl.6',
        data_dir7:'part.tbl.7',
        data_dir8:'part.tbl.8',
        data_dir9:'part.tbl.9',
        data_dir10:'part.tbl.10',
        data_dir11:'part.tbl.11',
        data_dir12:'part.tbl.12',
        data_dir13:'part.tbl.13',
        data_dir14:'part.tbl.14',
        data_dir15:'part.tbl.15',
        data_dir16:'part.tbl.16'
))
reject limit unlimited parallel;

drop table c_et;
create table c_et(
   c_custkey          number ,
   c_name             varchar(25) ,
   c_address          varchar(40) ,
   c_nationkey        number ,
   c_phone            char(15) ,
   c_acctbal          number ,
   c_mktsegment       char(10) ,
   c_comment          varchar(117)
)
organization external (
type ORACLE_LOADER
default directory data_dir1
access parameters
(
                        records delimited by newline
            nobadfile
            nologfile
                        fields terminated by '|'
                        missing field values are null
             )
             location (
        data_dir1:'customer.tbl.1',
        data_dir2:'customer.tbl.2',
        data_dir3:'customer.tbl.3',
        data_dir4:'customer.tbl.4',
        data_dir5:'customer.tbl.5',
        data_dir6:'customer.tbl.6',
        data_dir7:'customer.tbl.7',
        data_dir8:'customer.tbl.8',
        data_dir9:'customer.tbl.9',
        data_dir10:'customer.tbl.10',
        data_dir11:'customer.tbl.11',
        data_dir12:'customer.tbl.12',
        data_dir13:'customer.tbl.13',
        data_dir14:'customer.tbl.14',
        data_dir15:'customer.tbl.15',
        data_dir16:'customer.tbl.16'
))
reject limit unlimited parallel;

drop table s_et;
create table s_et(
   s_suppkey          number ,
```

```
   s_name          char(25) ,
   s_address       varchar(40) ,
   s_nationkey     number ,
   s_phone         char(15) ,
   s_acctbal       number ,
   s_comment       varchar(101)
)
organization external (
type ORACLE_LOADER
default directory data_dir1
access parameters
(
                    records delimited by newline
          nobadfile
          nologfile
          fields terminated by '|'
          missing field values are null
           )
           location (
        data_dir9:'supplier.tbl'
           ))
reject limit unlimited parallel;

drop table n_et ;
create table n_et(
   n_nationkey     number ,
   n_name          char(25) ,
   n_regionkey     number ,
   n_comment       varchar(152)
)
organization external (
type ORACLE_LOADER
default directory data_dir1
access parameters
(
                    records delimited by newline
          nobadfile
          nologfile
                    fields terminated by '|'
                    missing field values are null
           )
           location (
           data_dir9:'nation.tbl'))
reject limit unlimited;

drop table r_et;
create table r_et(
   r_regio nkey    number ,
   r_name          char(25) ,
   r_comment       varchar(152)
)
organization external (
type ORACLE_LOADER
default directory data_dir1
access parameters
(
                    records delimited by newline
          nobadfile
          nologfile
                    fields terminated by '|'
                    missing field values are null
           )
           location (
           data_dir9:'region.tbl'))
reject limit unlimited;

rem drop table lineitem;
create table lineitem(
   l_shipdate      ,
   l_orderkey      NOT NULL,
   l_discount      NOT NULL,
   l_extendedprice NOT NULL,
   l_suppkey       NOT NULL,
   l_quantity      NOT NULL,
   l_returnflag    ,
   l_partkey       NOT NULL,
   l_linestatus    ,
   l_tax           NOT NULL,
   l_commitdate    ,
   l_receiptdate   ,
   l_shipmode      ,
   l_linenumber    NOT NULL,
   l_shipinstruct  ,
   l_comment
)
pctfree 1
pctused 99
initrans 10
storage (initial 900m next 900m freelist groups 4 freelists 99)
parallel
nologging
partition by range (l_shipdate)
subpartition by hash(l_partkey)
subpartitions 16
(
partition item1 values less than (to_date('1992-01-01','YYYY-MM-
DD'))
store in (ts_l1,ts_l2,ts_l3,ts_l4,ts_l5,ts_l6,ts_l7,ts_l8)
,
partition item2 values less than (to_date('1992-02-01','YYYY-MM-
DD'))
store in (ts_l9,ts_l10,ts_l11,ts_l12,ts_l13,ts_l14,ts_l15,ts_l16)
,
partition item3 values less than (to_date('1992-03-01','YYYY-MM-
DD'))
store in (ts_l17,ts_l18,ts_l19,ts_l20,ts_l21,ts_l22,ts_l23,ts_l24)
,
partition item4 values less than (to_date('1992-04-01','YYYY-MM-
DD'))
store in (ts_l25,ts_l26,ts_l27,ts_l28,ts_l29,ts_l30,ts_l31,ts_l32)
,
partition item5 values less than (to_date('1992-05-01','YYYY-MM-
DD'))
store in (ts_l33,ts_l34,ts_l35,ts_l36,ts_ l37,ts_l38,ts_l39,ts_l40)
,
partition item6 values less than (to_date('1992-06-01','YYYY-MM-
DD'))
store in (ts_l41,ts_l42,ts_l43,ts_l44,ts_l45,ts_l46,ts_l47,ts_l48)
,
partition item7 values less than (to_date('1992-07-01','YYYY-MM-
DD'))
store in (ts_l49,ts_l50,ts_l51,ts_l52,ts_l53,ts_l54,ts_l55,ts_l56)
,
partition item8 values less than (to_date('1992-08-01','YYYY-MM-
DD'))
store in (ts_l57,ts_l58,ts_l59,ts_l60,ts_l61,ts_l62,ts_l63,ts_l64)
,
partition item9 values less than (to_date('1992-09-01','YYYY-MM-
DD'))
store in (ts_l65,ts_l66,ts_l67,ts_l68,ts_l69,ts_l70,ts_l71,ts_l72)
,
partition item10 values less than (to_date('1992-10-01','YYYY-MM-
DD'))
store in (ts_l73,ts_l74,ts_l75,ts_l76,ts_l77,ts_l78,ts_l79,ts_l80)
,
partition item11 values less than (to_date('1992-11-01','YYYY-MM-
DD'))
store in (ts_l81,ts_l82,ts_l83,ts_l84,ts_l85,ts_l86,ts_l87,ts_l88)
,
partition item12 values less than (to_date('1992-12-01','YYYY-MM-
DD'))
store in (ts_l89,ts_l90,ts_l91,ts_l92,ts_l93,ts_l94,ts_l95,ts_l96)
```

```
,
partition item13 values less than (to_date('1993-01-01','YYYY-MM-
DD'))
store in (ts_l97,ts_l98,ts_l99,ts_l100,ts_l101,ts_l102,ts_l103,ts_l104)
,
partition item14 values less than (to_date('1993-02-01','YYYY-MM-
DD'))
store in (ts_l105,ts_l106,ts_l107,ts_l108,ts_l109,ts_l110,ts_l111,ts_l112)
,
partition item15 values less than (to_date('1993-03-01','YYYY-MM-
DD'))
store in (ts_l113,ts_l114,ts_l115,ts_l116,ts_l117,ts_l118,ts_l119,ts_l120)
,
partition item16 values less than (to_date('1993-04-01','YYYY-MM-
DD'))
store in (ts_l121,ts_l122,ts_l123,ts_l124,ts_l125,ts_l126,ts_l127,ts_l128)
,
partition item17 values less than (to_date('1993-05-01','YYYY-MM-
DD'))
store in (ts_l129,ts_l130,ts_l131,ts_l132,ts_l133,ts_l134,ts_l135,ts_l136)
,
partition item18 values less than (to_date('1993-06-01','YYYY-MM-
DD'))
store in (ts_l137,ts_l138,ts_l139,ts_l140,ts_l141,ts_l142,ts_l143,ts_l144)
,
partition item19 values less than (to_date('1993-07-01','YYYY-MM-
DD'))
store in (ts_l145,ts_l146,ts_l147,ts_l148,ts_l149,ts_l150,ts_l151,ts_l152)
,
partition item20 values less than (to_date('1993-08-01','YYYY-MM-
DD'))
store in (ts_l153,ts_l154,ts_l155,ts_l156,ts_l157,ts_l158,ts_l159,ts_l160)
,
partition item21 values less than (to_date('1993-09-01','YYYY-MM-
DD'))
store in (ts_l161,ts_l162,ts_l163,ts_l164,ts_l165,ts_l166,ts_l167,ts_l168)
,
partition item22 values less than (to_date('1993-10-01','YYYY-MM-
DD'))
store in (ts_l169,ts_l170,ts_l171,ts_l172,ts_l173,ts_l174,ts_l175,ts_l176)
,
partition item23 values less than (to_date('1993-11-01','YYYY-MM-
DD'))
store in (ts_l177,ts_l178,ts_l179,ts_l180,ts_l181,ts_l182,ts_l183,ts_l184)
,
partition item24 values less than (to_date('1993-12-01','YYYY-MM-
DD'))
store in (ts_l185,ts_l186,ts_l187,ts_l188,ts_l189,ts_l190,ts_l191,ts_l192)
,
partition item25 values less than (to_date('1994-01-01','YYYY-MM-
DD'))
store in (ts_l193,ts_l194,ts_l195,ts_l196,ts_l197,ts_l198,ts_l199,ts_l200)
,
partition item26 values less than (to_date('1994-02-01','YYYY-MM-
DD'))
store in (ts_l201,ts_l202,ts_l203,ts_l204,ts_l205,ts_l206,ts_l207,ts_l208)
,
partition item27 values less than (to_date('1994-03-01','YYYY-MM-
DD'))
store in (ts_l209,ts_l210,ts_l211,ts_l212,ts_l213,ts_l214,ts_l215,ts_l216)
,
partition item28 values less than (to_date('1994-04-01','YYYY-MM-
DD'))
store in (ts_l217,ts_l218,ts_l219,ts_l220,ts_l221,ts_l222,ts_l223,ts_l224)
,
partition item29 values less than (to_date('1994-05-01','YYYY-MM-
DD'))
store in (ts_l225,ts_l226,ts_l227,ts_l228,ts_l229,ts_l230,ts_l231,ts_l232)
,
partition item30 values less than (to_date('1994-06-01','YYYY-MM-
DD'))
store in (ts_l233,ts_l234,ts_l235,ts_l236,ts_l237,ts_l238,ts_l239,ts_l240)
,
partition item31 values less than (to_date('1994-07-01','YYYY-MM-
DD'))
store in (ts_l241,ts_l242,ts_l243,ts_l244,ts_l245,ts_l246,ts_l247,ts_l248)
,
partition item32 values less than (to_date('1994-08-01','YYYY-MM-
DD'))
store in (ts_l249,ts_l250,ts_l251,ts_l252,ts_l253,ts_l254,ts_l255,ts_l256)
,
partition item33 values less than (to_date('1994-09-01','YYYY-MM-
DD'))
store in (ts_l257,ts_l258,ts_l259,ts_l260,ts_l261,ts_l262,ts_l263,ts_l264)
,
partition item34 values less than (to_date('1994-10-01','YYYY-MM-
DD'))
store in (ts_l265,ts_l266,ts_l267,ts_l268,ts_l269,ts_l270,ts_l271,ts_l272)
,
partition item35 values less than (to_date('1994-11-01','YYYY-MM-
DD'))
store in (ts_l273,ts_l274,ts_l275,ts_l276,ts_l277,ts_l278,ts_l279,ts_l280)
,
partition item36 values less than (to_date('1994-12-01','YYYY-MM-
DD'))
store in (ts_l281,ts_l282,ts_l283,ts_l284,ts_l285,ts_l286,ts_l287,ts_l288)
,
partition item37 values less than (to_date('1995-01-01','YYYY-MM-
DD'))
store in (ts_l289,ts_l290,ts_l291,ts_l292,ts_l293,ts_l294,ts_l295,ts_l296)
,
partition item38 values less than (to_date('1995-02-01','YYYY-MM-
DD'))
store in (ts_l297,ts_l298,ts_l299,ts_l300,ts_l301,ts_l302,ts_l303,ts_l304)
,
partition item39 values less than (to_date('1995-03-01','YYYY-MM-
DD'))
store in (ts_l305,ts_l306,ts_l307,ts_l308,ts_l309,ts_l310,ts_l311,ts_l312)
,
partition item40 values less than (to_date('1995-04-01','YYYY-MM-
DD'))
store in (ts_l313,ts_l314,ts_l315,ts_l316,ts_l317,ts_l318,ts_l319,ts_l320)
,
partition item41 values less than (to_date('1995-05-01','YYYY-MM-
DD'))
store in (ts_l321,ts_l322,ts_l323,ts_l324,ts_l325,ts_l326,ts_l327,ts_l328)
,
partition item42 values less than (to_date('1995-06-01','YYYY-MM-
DD'))
store in (ts_l329,ts_l330,ts_l331,ts_l332,ts_l333,ts_l334,ts_l335,ts_l336)
,
partition item43 values less than (to_date('1995-07-01','YYYY-MM-
DD'))
store in (ts_l337,ts_l338,ts_l339,ts_l340,ts_l341,ts_l342,ts_l343,ts_l344)
,
partition item44 values less than (to_date('1995-08-01','YYYY-MM-
DD'))
store in (ts_l345,ts_l346,ts_l347,ts_l348,ts_l349,ts_l350,ts_l351,ts_l352)
,
partition item45 values less than (to_date('1995-09-01','YYYY-MM-
DD'))
store in (ts_l353,ts_l354,ts_l355,ts_l356,ts_l357,ts_l358,ts_l359,ts_l360)
,
partition item46 values less than (to_date('1995-10-01','YYYY-MM-
DD'))
store in (ts_l361,ts_l362,ts_l363,ts_l364,ts_l365,ts_l366,ts_l367,ts_l368)
,
partition item47 values less than (to_date('1995-11-01','YYYY-MM-
DD'))
store in (ts_l369,ts_l370,ts_l371,ts_l372,ts_l373,ts_l374,ts_l375,ts_l376)
,
```

```
partition item48 values less than (to_date('1995-12-01','YYYY-MM-
DD'))
store in (ts_l377,ts_l378,ts_l379,ts_l380,ts_l381,ts_l382,ts_l383,ts_l384)
,
partition item49 values less than (to_date('1996-01-01','YYYY-MM-
DD'))
store in (ts_l385,ts_l386,ts_l387,ts_l388,ts_l389,ts_l390,ts_l391,ts_l392)
,
partition item50 values less than (to_date('1996-02-01','YYYY-MM-
DD'))
store in (ts_l393,ts_l394,ts_l395,ts_l396,ts_l397,ts_l398,ts_l399,ts_l400)
,
partition item51 values less than (to_date('1996-03-01','YYYY-MM-
DD'))
store in (ts_l401,ts_l402,ts_l403,ts_l404,ts_l405,ts_l406,ts_l407,ts_l408)
,
partition item52 values less than (to_date('1996-04-01','YYYY-MM-
DD'))
store in (ts_l409,ts_l410,ts_l411,ts_l412,ts_l413,ts_l414,ts_l415,ts_l416)
,
partition item53 values less than (to_date('1996-05-01','YYYY-MM-
DD'))
store in (ts_l417,ts_l418,ts_l419,ts_l420,ts_l421,ts_l422,ts_l423,ts_l424)
,
partition item54 values less than (to_date('1996-06-01','YYYY-MM-
DD'))
store in (ts_l425,ts_l426,ts_l427,ts_l428,ts_l429,ts_l430,ts_l431,ts_l432)
,
partition item55 values less than (to_date('1996-07-01','YYYY-MM-
DD'))
store in (ts_l433,ts_l434,ts_l435,ts_l436,ts_l437,ts_l438,ts_l439,ts_l440)
,
partition item56 values less than (to_date('1996-08-01','YYYY-MM-
DD'))
store in (ts_l441,ts_l442,ts_l443,ts_l444,ts_l445,ts_l446,ts_l447,ts_l448)
,
partition item57 values less than (to_date('1996-09-01','YYYY-MM-
DD'))
store in (ts_l449,ts_l450,ts_l451,ts_l452,ts_l453,ts_l454,ts_l455,ts_l456)
,
partition item58 values less than (to_date('1996-10-01','YYYY-MM-
DD'))
store in (ts_l457,ts_l458,ts_l459,ts_l460,ts_l461,ts_l462,ts_l463,ts_l464)
,
partition item59 values less than (to_date('1996-11-01','YYYY-MM-
DD'))
store in (ts_l465,ts_l466,ts_l467,ts_l468,ts_l469,ts_l470,ts_l471,ts_l472)
,
partition item60 values less than (to_date('1996-12-01','YYYY-MM-
DD'))
store in (ts_l473,ts_l474,ts_l475,ts_l476,ts_l477,ts_l478,ts_l479,ts_l480)
,
partition item61 values less than (to_date('1997-01-01','YYYY-MM-
DD'))
store in (ts_l481,ts_l482,ts_l483,ts_l484,ts_l485,ts_l486,ts_l487,ts_l488)
,
partition item62 values less than (to_date('1997-02-01','YYYY-MM-
DD'))
store in (ts_l489,ts_l490,ts_l491,ts_l492,ts_l493,ts_l494,ts_l495,ts_l496)
,
partition item63 values less than (to_date('1997-03-01','YYYY-MM-
DD'))
store in (ts_l497,ts_l498,ts_l499,ts_l500,ts_l501,ts_l502,ts_l503,ts_l504)
,
partition item64 values less than (to_date('1997-04-01','YYYY-MM-
DD'))
store in (ts_l505,ts_l506,ts_l507,ts_l508,ts_l509,ts_l510,ts_l511,ts_l512)
,
partition item65 values less than (to_date('1997-05-01','YYYY-MM-
DD'))
store in (ts_l513,ts_l514,ts_l515,ts_l516,ts_l517,ts_l518,ts_l519,ts_l520)
,
partition item66 values less than (to_date('1997-06-01','YYYY-MM-
DD'))
store in (ts_l521,ts_l522,ts_l523,ts_l524,ts_l525,ts_l526,ts_l527,ts_l528)
,
partition item67 values less than (to_date('1997-07-01','YYYY-MM-
DD'))
store in (ts_l529,ts_l530,ts_l531,ts_l532,ts_l533,ts_l534,ts_l535,ts_l536)
,
partition item68 values less than (to_date('1997-08-01','YYYY-MM-
DD'))
store in (ts_l537,ts_l538,ts_l539,ts_l540,ts_l541,ts_l542,ts_l543,ts_l544)
,
partition item69 values less than (to_date('1997-09-01','YYYY-MM-
DD'))
store in (ts_l545,ts_l546,ts_l547,ts_l548,ts_l549,ts_l550,ts_l551,ts_l552)
,
partition item70 values less than (to_date('1997-10-01','YYYY-MM-
DD'))
store in (ts_l553,ts_l554,ts_l555,ts_l556,ts_l557,ts_l558,ts_l559,ts_l560)
,
partition item71 values less than (to_date('1997-11-01','YYYY-MM-
DD'))
store in (ts_l561,ts_l562,ts_l563,ts_l564,ts_l565,ts_l566,ts_l567,ts_l568)
,
partition item72 values less than (to_date('1997-12-01','YYYY-MM-
DD'))
store in (ts_l569,ts_l570,ts_l571,ts_l572,ts_l573,ts_l574,ts_l575,ts_l576)
,
partition item73 values less than (to_date('1998-01-01','YYYY-MM-
DD'))
store in (ts_l577,ts_l578,ts_l579,ts_l580,ts_l581,ts_l582,ts_l583,ts_l584)
,
partition item74 values less than (to_date('1998-02-01','YYYY-MM-
DD'))
store in (ts_l585,ts_l586,ts_l587,ts_l588,ts_l589,ts_l590,ts_l591,ts_l592)
,
partition item75 values less than (to_date('1998-03-01','YYYY-MM-
DD'))
store in (ts_l593,ts_l594,ts_l595,ts_l596,ts_l597,ts_l598,ts_l599,ts_l600)
,
partition item76 values less than (to_date('1998-04-01','YYYY-MM-
DD'))
store in (ts_l601,ts_l602,ts_l603,ts_l604,ts_l605,ts_l606,ts_l607,ts_l608)
,
partition item77 values less than (to_date('1998-05-01','YYYY-MM-
DD'))
store in (ts_l609,ts_l610,ts_l611,ts_l612,ts_l613,ts_l614,ts_l615,ts_l616)
,
partition item78 values less than (to_date('1998-06-01','YYYY-MM-
DD'))
store in (ts_l617,ts_l618,ts_l619,ts_l620,ts_l621,ts_l622,ts_l623,ts_l624)
,
partition item79 values less than (to_date('1998-07-01','YYYY-MM-
DD'))
store in (ts_l625,ts_l626,ts_l627,ts_l628,ts_l629,ts_l630,ts_l631,ts_l632)
,
partition item80 values less than (to_date('1998-08-01','YYYY-MM-
DD'))
store in (ts_l633,ts_l634,ts_l635,ts_l636,ts_l637,ts_l638,ts_l639,ts_l640)
,
partition item81 values less than (to_date('1998-09-01','YYYY-MM-
DD'))
store in (ts_l641,ts_l642,ts_l643,ts_l644,ts_l645,ts_l646,ts_l647,ts_l648)
,
partition item82 values less than (to_date('1998-10-01','YYYY-MM-
DD'))
store in (ts_l649,ts_l650,ts_l651,ts_l652,ts_l653,ts_l654,ts_l655,ts_l656)
,
partition item83 values less than (to_date('1998-11-01','YYYY-MM-
DD'))
```

```
store in (ts_l657,ts_l658,ts_l659,ts_l660,ts_l661,ts_l662,ts_l663,ts_l664)
,
partition item84 values less than (MAXVALUE)
store in (ts_l665,ts_l666,ts_l667,ts_l668,ts_l669,ts_l670,ts_l671,ts_l672)
)
as select
    l_shipdate          ,
    l_orderkey          ,
    l_discount          ,
    l_extendedprice     ,
    l_suppkey           ,
    l_quantity          ,
    l_returnflag        ,
    l_partkey           ,
    l_linestatus        ,
    l_tax               ,
    l_commitdate        ,
    l_receiptdate       ,
    l_shipmode          ,
    l_linenumber        ,
    l_shipinstruct      ,
    l_comment
from l_et;


rem drop table orders;
create table orders(
    o_orderdate         ,
    o_orderkey          NOT NULL,
    o_custkey           NOT NULL,
    o_orderpriority     ,
    o_shippriority      ,
    o_clerk             ,
    o_orderstatus       ,
    o_totalprice        ,
    o_comment
)
pctfree 1
pctused 99
initrans 10
storage (initial 410m next 410m freelist groups 4 freelists 99)
parallel
nologging
partition by range (o_orderdate)
subpartition by hash(o_custkey)
subpartitions 16
(
partition ord1 values less than (to_date('1992-01-01','YYYY-MM-DD'))
store in (ts_o1,ts_o2,ts_o3,ts_o4,ts_o5,ts_o6,ts_o7,ts_o8)
,
partition ord2 values less than (to_date('1992-02-01','YYYY-MM-DD'))
store in (ts_o9,ts_o10,ts_o11,ts_o12,ts_o13,ts_o14,ts_o15,ts_o16)
,
partition ord3 values less than (to_date('1992-03-01','YYYY-MM-DD'))
store in (ts_o17,ts_o18,ts_o19,ts_o20,ts_o21,ts_o22,ts_o23,ts_o24)
,
partition ord4 values less than (to_date('1992-04-01','YYYY-MM-DD'))
store in (ts_o25,ts_o26,ts_o27,ts_o28,ts_o29,ts_o30,ts_o31,ts_o32)
,
partition ord5 values less than (to_date('1992-05-01','YYYY-MM-DD'))
store in (ts_o33,ts_o34,ts_o35,ts_o36,ts_o37,ts_o38,ts_o39,ts_o40)
,
partition ord6 values less than (to_date('1992-06-01','YYYY-MM-DD'))
store in (ts_o41,ts_o42,ts_o43,ts_o44,ts_o45,ts_o46,ts_o47,ts_o48)
,
partition ord7 values less than (to_date('1992-07-01','YYYY-MM-DD'))
store in (ts_o49,ts_o50,ts_o51,ts_o52,ts_o53,ts_o54,ts_o55,ts_o56)
,
partition ord8 values less than (to_date('1992-08-01','YYYY-MM-DD'))
store in (ts_o57,ts_o58,ts_o59,ts_o60,ts_o61,ts_o62,ts_o63,ts_o64)
,
partition ord9 values less than (to_date('1992-09-01','YYYY-MM-DD'))
store in (ts_o65,ts_o66,ts_o67,ts_o68,ts_o69,ts_o70,ts_o71,ts_o72)
,
partition ord10 values less than (to_date('1992-10-01','YYYY-MM-
DD'))
store in (ts_o73,ts_o74,ts_o75,ts_o76,ts_o77,ts_o78,ts_o79,ts_o80)
,
partition ord11 values less than (to_date('1992-11-01','YYYY-MM-
DD'))
store in (ts_o81,ts_o82,ts_o83,ts_o84,ts_o85,ts_o86,ts_o87,ts_o88)
,
partition ord12 values less than (to_date('1992-12-01','YYYY-MM-
DD'))
store in (ts_o89,ts_o90,ts_o91,ts_o92,ts_o93,ts_o94,ts_o95,ts_o96)
,
partition ord13 values less than (to_date('1993-01-01','YYYY-MM-
DD'))
store in
(ts_o97,ts_o98,ts_o99,ts_o100,ts_o101,ts_o102,ts_o103,ts_o104)
,
partition ord14 values less than (to_date('1993-02-01','YYYY-MM-
DD'))
store in
(ts_o105,ts_o106,ts_o107,ts_o108,ts_o109,ts_o110,ts_o111,ts_o112)
,
partition ord15 values less than (to_date('1993-03-01','YYYY-MM-
DD'))
store in
(ts_o113,ts_o114,ts_o115,ts_o116,ts_o117,ts_o118,ts_o119,ts_o120)
,
partition ord16 values less than (to_date('1993-04-01','YYYY-MM-
DD'))
store in
(ts_o121,ts_o122,ts_o123,ts_o124,ts_o125,ts_o126,ts_o127,ts_o128)
,
partition ord17 values less than  (to_date('1993-05-01','YYYY-MM-
DD'))
store in
(ts_o129,ts_o130,ts_o131,ts_o132,ts_o133,ts_o134,ts_o135,ts_o136)
,
partition ord18 values less than (to_date('1993-06-01','YYYY-MM-
DD'))
store in
(ts_o137,ts_o138,ts_o139,ts_o140,ts_o141,ts_o142,ts_o143,ts_o144)
,
partition ord19 values less than (to_date('1993-07-01','YYYY-MM-
DD'))
store in
(ts_o145,ts_o146,ts_o147,ts_o148,ts_o149,ts_o150,ts_o151,ts_o152)
,
partition ord20 values less than (to_date('1993-08-01','YYYY-MM-
DD'))
store in
(ts_o153,ts_o154,ts_o155,ts_o156,ts_o157,ts_o158,ts_o159,ts_o160)
,
partition ord21 values less than (to_date('1993-09-01','YYYY-MM-
DD'))
store in
(ts_o161,ts_o162,ts_o163,ts_o164,ts_o165,ts_o166,ts_o167,ts_o168)
,
partition ord22 values less than (to_date('1993-10-01','YYYY-MM-
DD'))
store in
(ts_o169,ts_o170,ts_o171,ts_o172,ts_o173,ts_o174,ts_o175,ts_o176)
,
partition ord23 values less than (to_date('1993-11-01','YYYY-MM-
DD'))
store in
(ts_o177,ts_o178,ts_o179,ts_o180,ts_o181,ts_o182,ts_o183,ts_o184)
,
```

```
partition ord24 values less than (to_date('1993-12-01','YYYY-MM-DD'))
store in (ts_o185,ts_o186,ts_o187,ts_o188,ts_o189,ts_o190,ts_o191,ts_o192)
,
partition ord25 values less than (to_date('1994-01-01','YYYY-MM-DD'))
store in (ts_o193,ts_o194,ts_o195,ts_o196,ts_o197,ts_o198,ts_o199,ts_o200)
,
partition ord26 values less than (to_date('1994-02-01','YYYY-MM-DD'))
store in (ts_o201,ts_o202,ts_o203,ts_o204,ts_o205,ts_o206,ts_o207,ts_o208)
,
partition ord27 values less than (to_date('1994-03-01','YYYY-MM-DD'))
store in (ts_o209,ts_o210,ts_o211,ts_o212,ts_o213,ts_o214,ts_o215,ts_o216)
,
partition ord28 values less than (to_date('1994-04-01','YYYY-MM-DD'))
store in (ts_o217,ts_o218,ts_o219,ts_o220,ts_o221,ts_o222,ts_o223,ts_o224)
,
partition ord29 values less than (to_date('1994-05-01','YYYY-MM-DD'))
store in (ts_o225,ts_o226,ts_o227,ts_o228,ts_o229,ts_o230,ts_o231,ts_o232)
,
partition ord30 values less than (to_date('1994-06-01','YYYY-MM-DD'))
store in (ts_o233,ts_o234,ts_o235,ts_o236,ts_o237,ts_o238,ts_o239,ts_o240)
,
partition ord31 values less than (to_date('1994-07-01','YYYY-MM-DD'))
store in (ts_o241,ts_o242,ts_o243,ts_o244,ts_o245,ts_o246,ts_o247,ts_o248)
,
partition ord32 values less than (to_date('1994-08-01','YYYY-MM-DD'))
store in (ts_o249,ts_o250,ts_o251,ts_o252,ts_o253,ts_o254,ts_o255,ts_o256)
,
partition ord33 values less than (to_date('1994-09-01','YYYY-MM-DD'))
store in (ts_o257,ts_o258,ts_o259,ts_o260,ts_o261,ts_o262,ts_o263,ts_o264)
,
partition ord34 values less than (to_date('1994-10-01','YYYY-MM-DD'))
store in (ts_o265,ts_o266,ts_o267,ts_o268,ts_o269,ts_o270,ts_o271,ts_o272)
,
partition ord35 values less than (to_date('1994-11-01','YYYY-MM-DD'))
store in (ts_o273,ts_o274,ts_o275,ts_o276,ts_o277,ts_o278,ts_o279,ts_o280)
,
partition ord36 values less than (to_date('1994-12-01','YYYY-MM-DD'))
store in (ts_o281,ts_o282,ts_o283,ts_o284,ts_o285,ts_o286,ts_o287,ts_o288)
,
partition ord37 values less than (to_date('1995-01-01','YYYY-MM-DD'))
store in (ts_o289,ts_o290,ts_o291,ts_o292,ts_o293,ts_o294,ts_o295,ts_o296)
,
partition ord38 values less than (to_date('1995-02-01','YYYY-MM-DD'))
store in (ts_o297,ts_o298,ts_o299,ts_o300,ts_o301,ts_o302,ts_o303,ts_o304)
,
partition ord39 values less than (to_date('1995-03-01','YYYY-MM-DD'))
store in (ts_o305,ts_o306,ts_o307,ts_o308,ts_o309,ts_o310,ts_o311,ts_o312)
,
partition ord40 values less than (to_date('1995-04-01','YYYY-MM-DD'))
store in (ts_o313,ts_o314,ts_o315,ts_o316,ts_o317,ts_o318,ts_o319,ts_o320)
,
partition ord41 values less than (to_date('1995-05-01','YYYY-MM-DD'))
store in (ts_o321,ts_o322,ts_o323,ts_o324,ts_o325,ts_o326,ts_o327,ts_o328)
,
partition ord42 values less than (to_date('1995-06-01','YYYY-MM-DD'))
store in (ts_o329,ts_o330,ts_o331,ts_o332,ts_o333,ts_o334,ts_o335,ts_o336)
,
partition ord43 values less than (to_date('1995-07-01','YYYY-MM-DD'))
store in (ts_o337,ts_o338,ts_o339,ts_o340,ts_o341,ts_o342,ts_o343,ts_o344)
,
partition ord44 values less than (to_date('1995-08-01','YYYY-MM-DD'))
store in (ts_o345,ts_o346,ts_o347,ts_o348,ts_o349,ts_o350,ts_o351,ts_o352)
,
partition ord45 values less than (to_date('1995-09-01','YYYY-MM-DD'))
store in (ts_o353,ts_o354,ts_o355,ts_o356,ts_o357,ts_o358,ts_o359,ts_o360)
,
partition ord46 values less than (to_date('1995-10-01','YYYY-MM-DD'))
store in (ts_o361,ts_o362,ts_o363,ts_o364,ts_o365,t s_o366,ts_o367,ts_o368)
,
partition ord47 values less than (to_date('1995-11-01','YYYY-MM-DD'))
store in (ts_o369,ts_o370,ts_o371,ts_o372,ts_o373,ts_o374,ts_o375,ts_o376)
,
partition ord48 values less than (to_date('1995-12-01','YYYY-MM-DD'))
store in (ts_o377,ts_o378,ts_o379,ts_o380,ts_o381,ts_o382,ts_o383,ts_o384)
,
partition ord49 values less than (to_date('1996-01-01','YYYY-MM-DD'))
store in (ts_o385,ts_o386,ts_o387,ts_o388,ts_o389,ts_o390,ts_o391,ts_o392)
,
partition ord50 values less than (to_date('1996-02-01','YYYY-MM-DD'))
store in (ts_o393,ts_o394,ts_o395,ts_o396,ts_o397,ts_o398,ts_o399,ts_o400)
,
partition ord51 values less than (to_date('1996-03-01','YYYY-MM-DD'))
store in (ts_o401,ts_o402,ts_o403,ts_o404,ts_o405,ts_o406,ts_o407,ts_o408)
,
```

```
partition ord52 values less than (to_date('1996-04-01','YYYY-MM-
DD'))
store in
(ts_o409,ts_o410,ts_o411,ts_o412,ts_o413,ts_o414,ts_o415,ts_o416)
,
partition ord53 values less than (to_date('1996-05-01','YYYY-MM-
DD'))
store in
(ts_o417,ts_o418,ts_o419,ts_o420,ts_o421,ts_o422,ts_o423,ts_o424)
,
partition ord54 values less than (to_date('1996-06-01','YYYY-MM-
DD'))
store in
(ts_o425,ts_o426,ts_o427,ts_o428,ts_o429,ts_o430,ts_o431,ts_o432)
,
partition ord55 values less than (to_date('1996-07-01','YYYY-MM-
DD'))
store in
(ts_o433,ts_o434,ts_o435,ts_o436,ts_o437,ts_o438,ts_o439,ts_o440)
,
partition ord56 values less than (to_date('1996-08-01','YYYY-MM-
DD'))
store in
(ts_o441,ts_o442,ts_o443,ts_o444,ts_o445,ts_o446,ts_o447,ts_o448)
,
partition ord57 values less than (to_date('1996-09-01','YYYY-MM-
DD'))
store in
(ts_o449,ts_o450,ts_o451,ts_o452,ts_o453,ts_o454,ts_o455,ts_o456)
,
partition ord58 values less than (to_date('1996-10-01','YYYY-MM-
DD'))
store in
(ts_o457,ts_o458,ts_o459,ts_o460,ts_o461,ts_o462,ts_o463,ts_o464)
,
partition ord59 values less than (to_date('1996-11-01','YYYY-MM-
DD'))
store in
(ts_o465,ts_o466,ts_o467,ts_o468,ts_o469,ts_o470,ts_o471,ts_o472)
,
partition ord60 values less than (to_date('1996-12-01','YYYY-MM-
DD'))
store in
(ts_o473,ts_o474,ts_o475,ts_o476,ts_o477,ts_o478,ts_o479,ts_o480)
,
partition ord61 values less than (to_date('1997-01-01','YYYY-MM-
DD'))
store in
(ts_o481,ts_o482,ts_o483,ts_o484,ts_o485,ts_o486,ts_o487,ts_o488)
,
partition ord62 values less than (to_date('1997-02-01','YYYY-MM-
DD'))
store in
(ts_o489,ts_o490,ts_o491,ts_o492,ts_o493,ts_o494,ts_o495,ts_o496)
,
partition ord63 values less than (to_date('1997-03-01','YYYY-MM-
DD'))
store in
(ts_o497,ts_o498,ts_o499,ts_o500,ts_o501,ts_o502,ts_o503,ts_o504)
,
partition ord64 values less than (to_date('1997-04-01','YYYY-MM-
DD'))
store in
(ts_o505,ts_o506,ts_o507,ts_o508,ts_o509,ts_o510,ts_o511,ts_o512)
,
partition ord65 values less than (to_date('1997-05-01','YYYY-MM-
DD'))
store in
(ts_o513,ts_o514,ts_o515,ts_o516,ts_o517,ts_o518,ts_o519,ts_o520)
,
partition ord66 values less than (to_date('1997-06-01','YYYY-MM-
DD'))
store in
(ts_o521,ts_o522,ts_o523,ts_o524,ts_o525,ts_o526,ts_o527,ts_o528)
,
partition ord67 values less than (to_date('1997-07-01','YYYY-MM-
DD'))
store in
(ts_o529,ts_o530,ts_o531,ts_o532,ts_o533,ts_o534,ts_o535,ts_o536)
,
partition ord68 values less than (to_date('1997-08-01','YYYY-MM-
DD'))
store in
(ts_o537,ts_o538,ts_o539,ts_o540,ts_o541,ts_o542,ts_o543,ts_o544)
,
partition ord69 values less than (to_date('1997-09-01','YYYY-MM-
DD'))
store in
(ts_o545,ts_o546,ts_o547,ts_o548,ts_o549,ts_o550,ts_o551,ts_o552)
,
partition ord70 values less than (to_date('1997-10-01','YYYY-MM-
DD'))
store in
(ts_o553,ts_o554,ts_o555,ts_o556,ts_o557,ts_o558,ts_o559,ts_o560)
,
partition ord71 values less than (to_date('1997-11-01','YYYY-MM-
DD'))
store in
(ts_o561,ts_o562,ts_o563,ts_o564,ts_o565,ts_o566,ts_o567,ts_o568)
,
partition ord72 values less than (to_date('1997-12-01','YYYY-MM-
DD'))
store in
(ts_o569,ts_o570,ts_o571,ts_o572,ts_o573,ts_o574,ts_o575,ts_ o576)
,
partition ord73 values less than (to_date('1998-01-01','YYYY-MM-
DD'))
store in
(ts_o577,ts_o578,ts_o579,ts_o580,ts_o581,ts_o582,ts_o583,ts_o584)
,
partition ord74 values less than (to_date('1998-02-01','YYYY-MM-
DD'))
store in
(ts_o585,ts_o586,ts_o587,ts_o588,ts_o589,ts_o590,ts_o591,ts_o592)
,
partition ord75 values less than (to_date('1998-03-01','YYYY-MM-
DD'))
store in
(ts_o593,ts_o594,ts_o595,ts_o596,ts_o597,ts_o598,ts_o599,ts_o600)
,
partition ord76 values less than (to_date('1998-04-01','YYYY-MM-
DD'))
store in
(ts_o601,ts_o602,ts_o603,ts_o604,ts_o605,ts_o606,ts_o607,ts_o608)
,
partition ord77 values less than (to_date('1998-05-01','YYYY-MM-
DD'))
store in
(ts_o609,ts_o610,ts_o611,ts_o612,ts_o613,ts_o614,ts_o615,ts_o616)
,
partition ord78 values less than (to_date('1998-06-01','YYYY-MM-
DD'))
store in
(ts_o617,ts_o618,ts_o619,ts_o620,ts_o621,ts_o622,ts_o623,ts_o624)
,
partition ord79 values less than (to_date('1998-07-01','YYYY-MM-
DD'))
store in
(ts_o625,ts_o626,ts_o627,ts_o628,ts_o629,ts_o630,ts_o631,ts_o632)
,
```

```
partition ord80 values less than (to_date('1998-08-01','YYYY-MM-
DD'))
store in
(ts_o633,ts_o634,ts_o635,ts_o636,ts_o637,ts_o638,ts_o639,ts_o640)
,
partition ord81 values less than (to_date('1998-09-01','YYYY-MM-
DD'))
store in
(ts_o641,ts_o642,ts_o643,ts_o644,ts_o645,ts_o646,ts_o647,ts_o648)
,
partition ord82 values less than (to_date('1998-10-01','YYYY-MM-
DD'))
store in
(ts_o649,ts_o650,ts_o651,ts_o652,ts_o653,ts_o654,ts_o655,ts_o656)
,
partition ord83 values less than (to_date('1998-11-01','YYYY-MM-
DD'))
store in
(ts_o657,ts_o658,ts_o659,ts_o660,ts_o661,ts_o662,ts_o663,ts_o664)
,
partition ord84 values less than (MAXVALUE)
store in
(ts_o665,ts_o666,ts_o667,ts_o668,ts_o669,ts_o670,ts_o671,ts_o672) )
as select
   o_orderdate        ,
   o_orderkey         ,
   o_custkey          ,
   o_orderpriority    ,
   o_shippriority     ,
   o_clerk            ,
   o_orderstatus      ,
   o_totalprice       ,
   o_comment
from o_et;


rem 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
partition by hash(ps_partkey)
partitions 256
storage (initial 600m next 600m)
parallel
nologging
pctthreshold 50
tablespace ts_psupp1
as select
   ps_partkey         ,
   ps_suppkey         ,
   ps_supplycost      ,
   ps_availqty        ,
   ps_comment
from ps_et;


rem drop table customer;
create table customer(
   c_custkey          NOT NULL,
   c_mktsegment       ,
   c_nationkey        ,
   c_name             ,
   c_address          ,
   c_phone            ,
   c_acctbal          ,
   c_comment
)
pctfree 0
pctused 99
parallel
nologging
storage (initial 130m next 130m)
partition by hash (c_custkey)
partitions 256
store in
(ts_small1,ts_small2,ts_small3,ts_small4,ts_small5,ts_small6,ts_small7,t
s_small8,ts_small9,ts_small10,ts_small11,ts_small12,ts_small13,ts_sma
ll14,ts_small15,ts_small16,ts_small17,ts_small18,ts_small19,ts_small20
,ts_small21,ts_small22,ts_small23,ts_small24,ts_small25,ts_small26,ts_
small27,ts_small28,ts_small29,ts_small30,ts_small31,ts_small32)
as select
   c_custkey          ,
   c_mktsegment       ,
   c_nationkey        ,
   c_name             ,
   c_address          ,
   c_phone            ,
   c_acctbal          ,
   c_comment
from c_et;


rem drop table part;

create table part(
   p_partkey          NOT NULL,
   p_type             ,
   p_size             ,
   p_brand            ,
   p_name             ,
   p_container        ,
   p_mfgr             ,
   p_retailprice      ,
   p_comment
)
pctfree 0
pctused 99
parallel
nologging
storage (initial 150m next 150m)
partition by hash (p_partkey)
partitions 256
store in
(ts_small1,ts_small2,ts_small3,ts_small4,ts_small5,ts_small6,ts_small7,t
s_small8,ts_small9,ts_small10,ts_small11,ts_small12,ts_small13,ts_sma
ll14,ts_small15,ts_small16,ts_small17,ts_small18,ts_small19,ts_small20
,ts_small21,ts_small22,ts_small23,ts_small24,ts_small25,ts_small26,ts_
small27,ts_small28,ts_small29,ts_small30,ts_small31,ts_small32)
as select
   p_partkey          ,
   p_type             ,
   p_size             ,
   p_brand            ,
   p_name             ,
   p_container        ,
   p_mfgr             ,
   p_retailprice      ,
   p_comment
from p_et;


rem drop table supplier;
create table supplier(
   s_suppkey          NOT NULL,
   s_nationkey        ,
   s_comment          ,
```

```
   s_name          ,
   s_address       ,
   s_phone         ,
   s_acctbal
)
pctfree 0
pctused 99
parallel
nologging
storage (initial 60m next 10m)
partition by hash (s_suppkey)
partitions 256
store in
(ts_small1,ts_small2,ts_small3,ts_small4,ts_small5,ts_small6,ts_small7,t
s_small8,ts_small9,ts_small10,ts_small11,ts_small12,ts_small13,ts_sma
ll14,ts_small15,ts_small16,ts_small17,ts_small18,ts_small19,ts_small20
,ts_small21,ts_small22,ts_small23,ts_small24,ts_small25,ts_small26,ts_
small27,ts_small28,ts_small29,ts_small30,ts_small31,ts_small32)
as select
   s_suppkey       ,
   s_nationkey     ,
   s_comment       ,
   s_name          ,
   s_address       ,
   s_phone         ,
   s_acctbal
from s_et;

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

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


drop table l_et;
drop table o_et;
drop table ps_et;
drop table p_et;
drop table c_et;
drop table s_et;
drop table n_et;
drop table r_et;

!

echo DONE TABLE CREATION at `date`
```

B.4 ixcre.sh

```
#!/bin/ksh

echo START INDEX at `date`
sqlplus  tpch/tpch <<!
set echo on
set timing on
set termout on

rem drop index i_l_orderkey;
create index i_l_orderkey
on lineitem (l_orderkey)
global partition by hash (l_orderkey)
partitions 128
pctfree 5
initrans 10
tablespace ts_lokey
storage (freelist groups 4 freelists 99)
parallel
compute statistics
nologging;


rem drop index i_o_orderkey;
create unique index i_o_orderkey
on orders (o_orderkey)
global partition by hash (o_orderkey)
partitions 128
pctfree 2
initrans 10
tablespace ts_okey
storage (freelist groups 4 freelists 99 )
parallel
compute statistics
nologging;


rem drop index i_c_custkey;
create unique index i_c_custkey
on customer (c_custkey)
pctfree 2
initrans 10
tablespace ts_custkey
storage (freelists 99)
parallel
compute statistics
nologging;
!

echo DONE INDEX at `date`
```

B.5 anl.sh

```
#!/bin/ksh

echo START ANALYZE at `date`

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

execute dbms_stats.gather_schema_stats('TPCH' ,
estimate_percent => 1, degree => 128 ,
granularity => 'GLOBAL', method_opt => 'for all
columns size 1' );
connect / as sysdba
execute dbms_stats.gather_system_stats;
exec dbms_scheduler.disable('GATHER_STATS_JOB');
alter system switch logfile;
!

echo END ANALYZE at `date`
```

Appendix C Acid Scripts

C.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 RESULT
from lineitem
where l_orderkey = &&1;

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

exit;
```

C.2 a_query2.sql

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

set serverout on;

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

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

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

exit;
```

C.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`..."
```

C.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 dobatrans;
END;
/

exit;
```

C.5 atranspl.c

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

/*

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

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

   NOTES
     <other useful comments, qualifications, etc.>

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

*/

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

#include "atranspl.h"

/* Declare error handling functions */

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

/* declarations for ORDERS */

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

/* declarations for LINEITEM */

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

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

sb2 l_npricei;

/* other declarations */

int delta = 0;
double rprice;
double cost;

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

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

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

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

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

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

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

time_t curr_time;        /* Current Time                      */

/* OCI handles */

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

/* OCI bind handles */

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

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

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

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

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

char sqlstmt[1024];

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

void usage()
{

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

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

}


void ACIDexit() {

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

}



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

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

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

```
    if (type)
     (void) OCIErrorGet(errhp,1,NULL, (sb4 *) &errcode, (text*) msg,
                        2048,OCI_HTYPE_ERROR);
    else
     (void) OCIErrorGet(errhp,1,NULL, (sb4 *) &errcode, (text*) msg,
                        2048,OCI_HTYPE_ENV);
    fprintf(stderr,"%s\n",msg);
    break;
  }
 /* Rollback just in case */

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

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

  ACIDexit();

  exit(1);
}

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

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

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

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

  /* argv[1] – Process Number */

  proc_no = atoi(argv[1]);

  /* argv[2] – Number of Streams */

  num_streams = atoi(argv[2]);

  /* argv[3] – Commit? */

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

  /* argv[4] – Delta? */

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

  /* Process  optional parameters */

  argc -= 4;
  argv += 4;

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

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

  /* Initialize the cursors etc. */

  (void) ACIDinit();

  /* sleep for some time (triggering) */

  sleep(trig);

  /* start doing the ACID transactions */

  tr_start = gettime();

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

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

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

    /* Obtain l_key from l_key query */

    OCIsexec(tpcsvc,curi,errhp,1);

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

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

    /* Generate delta if necessary */

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

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

    curr_time = time(NULL);

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

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

    OCIsexec(tpcsvc,curr,errhp,1);

    curr_time = time(NULL);

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

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

    /* Shall we commit? */

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

  /* Enable session parallel dml */

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

  /* Enable session parallel ddl */

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


  /* Make session serializable */

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

  /* Set optimizer_index_cost_adj = 25 */

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

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


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

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



OCIbbname(curi,&l_keyi_bp,errhp,":l_key",ADR(l_key),SIZ(l_key),SQ
LT_INT);

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

#endif /* NOLKEY */

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

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

  /* bind variables */


OCIbbname(curr,l_key_bp,errhp,":l_key",ADR(l_key),SIZ(l_key),SQL
T_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),
SQLT_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_
FLT);

OCIbbname(curr,l_disc_bp,errhp,":l_disc",ADR(l_disc),SIZ(l_disc),SQ
LT_FLT);

OCIbbname(curr,l_eprice_bp,errhp,":l_eprice",ADR(l_eprice),SIZ(l_epr
ice),
        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_tp
rice),
        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_newep
rice),
        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),S
QLT_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);

}
```

C.6 atranspl.h

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

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

   DESCRIPTION

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

*/
#ifndef ATRANSPL_H

#define ATRANSPL_H

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

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

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

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

extern int errno;

#ifndef NULL
#define NULL 0
#endif

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

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#ifndef ub1
#define ub1 unsigned char
#endif

#define UNAME_LEN 64
```

```
#define WRITE_BUF_LEN 1024

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

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

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

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

#defin e 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_DE
FAULT)) != OCI_SUCCESS) \
      sql_error(errh,status,1); \
   else \
      DISCARD 0

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

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

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

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

#define ISOTXT "alter session set isolation_level = serializable"
#define PDMLTXT "alter session force parallel dml parallel (degree 4)"
#define PDDLTXT "alter session force parallel ddl parallel (degree 4)"
#define OICATXT "alter session set optimizer_index_cost_adj=25"

#define SQLTXT1 "BEGIN SELECT /*+ index(lineitem,i_l_orderkey)
*/ MAX(l_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 */
```

C.7 ckpt.sh

```
#!/bin/ksh
#
# $Header: ckpt.sh 08-aug-99.17:37:07 mpoess Exp $
#
# ckpt.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      ckpt.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#      Usage: ckpt.sh
#      Start database checkpoint
```

```
#
#   NOTES
#     <other useful comments, qualifications, etc.>
#
#   MODIFIED   (MM/DD/YY)
#   mpoess      08/08/99 - Creation
#   mpoess      08/08/99 - Creation
#

. $KIT_DIR/env

sqlplus -s /NOLOG << !

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

C.8 cnt_hist.sql

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

C.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 @/dbms/oracle10g/kit/acid/consistency/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 @/dbms/oracle10g/kit/acid/consistency/consist $j >>
$CON2
   echo "------------------------------" >> $CON2
done
   i=`expr $i + 1`
done
```

C.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
```

C.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
```

C.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;
```

C.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=/dbms/oracle10i/kit_acid
. $KIT_DIR/env

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

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

# get history count

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

# perform the consistency

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

i=0
```

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

cat $ORACLE_HOME/rdbms/log/alert_1g.log >
${DURA_DIR}/alert_1g.log.post_dura 2>&1
```

C.14 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=remsh

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

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

USER=$DATABASE_USER
PROG=atranspl


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

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

usage() {

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

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

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


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

# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

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

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

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

sleep 1

# start ACID transaction, Sleep for 60 second before COMMIT

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

# let's sleep 10 seconds before starting ACID query

sleep 10

# start ACID query with the same OKEY

echo "Running ACID query 10 seconds AFTER the start of ACID
Transaction" \
>> $TXN2FILE
echo "`date`" >> $TXN2FILE
if [ "$HOST" != "" ]
then
echo "Starting ACID query on node $HOST" >> $TXN2FILE
${RSH} ${HOST} -n ". /dbms/oracle10i/.profile;  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
```

C.15 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=remsh

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

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

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

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

echo "Running ACID query BEFORE the start of Isolation Test 2" >>
$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`" >> $TXN2FILE
if [ "$HOST" != "" ]
then
echo "Starting ACID query on node $HOST" >> $TXN2FILE
${RSH} ${HOST}  -n ". /dbms/oracle10i/.profile; sqlplus "$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
```

C.16 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=remsh

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

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE
if [ "$HOST" != "" ]
then
  rcp $KEYFILE ${HOST}:$KEYFILE
fi

sleep 1

# start ACID transaction, Sleep for 30 second before COMMIT

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

# let's sleep 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 [ "$HOST" != "" ]
then
echo "Starting TXN2 on node $HOST" >> $TXN2FILE
${RSH} ${HOST} ". /dbms/oracle10i/.profile; .
/dbms/oracle10i/kit_acid/env; $PROG 2 2 1 1 i$KEYFILE u$USER s1
b1" >> $TXN2FILE &
else
$PROG 2 2 1 1 i$KEYFILE u$USER s1 b1 >> $TXN2FILE &
fi

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

C.17 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=remsh

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

```
OUT_DIR=$ACID_OUT

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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


# generate key files

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

sleep 1

# start ACID transaction, Sleep for 30 second before ROLLBACK

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

# let's sleep 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 [ "$HOST" != "" ]
then
echo "Starting TXN2 on node $HOST" >> $TXN2FILE
${RSH} ${HOST} -n ". /dbms/oracle10i/.profile; .
/dbms/oracle10i/kit_acid/env; $PROG 2 2 1 1 i$KEYFILE u$USER s1
b1" >> $TXN2FILE &
else
$PROG 2 2 1 1 i$KEYFILE u$USER s1 b1 >> $TXN2FILE &
fi

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

C.18 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=remsh

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

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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

```
  --) break;;
  esac
  shift;
done


# generate key files

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

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

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

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

sleep 1

# start ACID transaction, Sleep for 60 second before COMMIT

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

# let's sleep 5 seconds before starting PARTSUPP query

sleep 5

# First generate PS_PARTKEY and PS_SUPPKEY

PSKEY=`randpsup 1`

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

if [ "$HOST" != "" ]
then
echo "Starting PARTSUPP query on node $HOST" >> $TXN2FILE
${RSH} ${HOST} -n ". /dbms/oracle10i/.profile; 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
```

C.19 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=remsh

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

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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

```
# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

if [ "$HOST" != "" ]
then
  rcp $KEYFILE ${HOST}:$KEYFILE
fi

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

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

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

sleep 1

# start Query 1, use 0 as the delta

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

# sleep 2 seconds before starting ACID transaction

sleep 2

# start ACID transaction, COMMIT after one second

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

if [ "$HOST" != "" ]
then
echo "Starting ACID transaction on node $HOST" >> $TXN2FILE
${RSH} ${HOST}  -n ". /dbms/oracle10i/.profile; .
/dbms/oracle10i/kit_acid/env; $PROG 1 1 1 0 i$KEYFILE u$USER s1"
>> $TXN2FILE &
else
$PROG 1 1 1 0 i$KEYFILE u$USER s1 >> $TXN2FILE &
fi

# start Query 1

sleep 2

echo "Running 2nd Query 21 at `date`" >> $TXN3FILE
sqlplus $USER @$KIT_DIR/acid/isolation/q21 >> $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
```

C.20 randkey.c

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

/*

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

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

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

#define ORDERCNT 150000.0

/* MK_SPARSE adopted from dss.h */

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

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

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

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

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

/* OCI handles */

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

OCIBind *l_key_bp;
OCIBind *o_key_bp;

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

char sqlstmt[1024];


void ACIDexit() {
  OCILogoff(tpcsvc,errhp);
  OCIhfree(tpcenv,OCI_HTYPE_STMT);
  OCIhfree(tpcsvc,OCI_HTYPE_SVCCTX);
```

```
  OCIhfree(tpcsrv,OCI_HTYPE_SERVER);
  OCIhfree(tpcusr,OCI_HTYPE_SESSION);
}


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

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

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

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

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

  ACIDexit();

  exit(1);
}


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

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

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

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

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

  argc -= 2;
  argv += 2;

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

  ACIDinit();

  /* initialize array for random numbers */

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

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

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

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

    /* Obtain l_key from l_key query */

    OCIsexec(tpcsvc,curi,errhp,1);

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

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

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

  ACIDexit();
  free(res);

}


void usage() {
```

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

void ACIDinit()
{

 /* run random seed */

 srand48(getpid());

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

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

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

 /* get username and password */

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

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


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

OCIaset(tpcusr,OCI_HTYPE_SESSION,lname,strlen(lname),OCI_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_SESSIO
N,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),SQL
T_INT);

OCIbbname(curi,o_key_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),SQ
LT_INT);
}
```

C.21 randpsup.c

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

/*

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

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

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

*/

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

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

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

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

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


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

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

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

 /* seed the random number generator */

 srand48(getpid());
```

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

  PART_SUPP_BRIDGE(ps_skey, pkey, supp);

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

  exit(0);
}


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

C.22 sample.sh

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

# $1 durability output file

. $KIT_DIR/env

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



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

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

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

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

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

C.23 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;
```

C.24 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     02/13/0 1 - 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;
```

C.25 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=/dbms/oracle10i/kit_acid/
. $KIT_DIR/env

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

usage() {

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

RSH=remsh
ITER=2000
STEM=${NUM_STREAMS}
let STEM="$STEM + 1" # add one for the update stream
SF=1
PROG=atranspl
IN=${ACID_DIR}/acid_in
DURA_DIR=$ACID_OUT/dura
OUT=$DURA_DIR/drate
DURA=$DURA_DIR/dura
KEY=${DURA_DIR}/key$$_
echo "$$" > ${DURA_DIR}/shellpid
USER=tpch/tpch
TRIG=1
HCNT=duracntb

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

# get all the options

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

#collect system info before durability start
cat /var/adm/syslog/syslog.log > ${DURA_DIR}/syslog_pre_dura 2>&1
ps -ef > ${DURA_DIR}/ps.out.pre_dura 2>&1
cat $ORACLE_HOME/rdbms/log/alert_1g.log >
${DURA_DIR}/alert_1g.log.pre_dura 2>&1

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
remote=""
while [ $i -lt $STEM ]
do
  if [ "X$HOST" != "X" -a "X$remote" != "X" ]
  then
    rcp ${KEY}${i} ${HOST}:${KEY}${i}
    ${RSH} ${HOST}  -n ". /dbms/oracle10i/.profile; .
/dbms/oracle10i/kit_acid/env; $PROG $i $STEM 1 0 i${KEY}${i}
o${OUT}${i} d${DURA}${i} u$USER s1"  &
    remote=""
  else
    $PROG $i $STEM 1 0 i${KEY}${i} o${OUT}${i} d${DURA}${i}
u$USER s1 &
    remote="y"
  fi
  T=`expr $T - $TRIG`
  i=`expr $i + 1`

done

wait
echo "ACID run completed"
```

C.26 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 @d_hist
sqlplus $DATABASE_USER @atrans
```

Appendix D Query text and Output

D.1 qryqual

```
-- TPC-H/TPC-R Pricing Summary Report Query (Q1)
-- Functional Query Definition
-- Approved February 1998


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

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


4 rows processed.
Query Processed in 5.03 seconds.
```

```
-- @(#)2.sql            2.1.6.2
-- TPC-H/TPC-R Minimum Cost Supplier Query (Q2)
-- Functional Query Definition
-- Approved February 1998


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

```
S_ACCTBAL            S_NAME               N_NAME
P_PARTKEY            P_MFGR
S_ADDRESS                         S_PHONE
S_COMMENT
9938.53              Supplier#000005359   UNITED KINGDOM
185358.00            Manufacturer#4
QKuHYh,vZGiwu2FWEJoLDx04              33-429-790-6131
blithely silent pinto beans are furiously. slyly final deposits acros
9937.84              Supplier#000005969   ROMANIA
108438.00            Manufacturer#1
ANDENSOSmk,miq23Xfb5RWt6dvUcvt6Qa     29-520-692-3537
carefully slow deposits use furiously. slyly ironic platelets above the
ironic
9936.22              Supplier#000005250   UNITED KINGDOM
249.00               Manufacturer#4
B3rqp0xbSEim4Mpy2RH J                 33-320-228-2957
blithely special packages are. stealthily express deposits across the
closely final instructi
9923.77              Supplier#000002324   GERMANY
29821.00             Manufacturer#4
y3OD9UywSTOk                          17-779-299-1839
quickly express packages breach quiet pinto beans. requ
9871.22              Supplier#000006373   GERMANY
43868.00             Manufacturer#5
J8fcXWsTqM                            17-813-485-8637
never silent deposits integrate furiously blit
9870.78              Supplier#000001286   GERMANY
81285.00             Manufacturer#2
YKA,E2fjiVd7eUrzp2Ef8j1QxGo2DFnosaTEH  17-516-924-4574
final theodolites cajole slyly special,
9870.78              Supplier#000001286   GERMANY
181285.00            Manufacturer#4
```

YKA,E2fjiVd7eUrzp2Ef8j1QxGo2DFnosaTEH 17-516-924-4574
final theodolites cajole slyly special,
9852.52 Supplier#000008973 RUSSIA
18972.00 Manufacturer#2
t5L67YdBYYH6o,Vz24jpDyQ9 32-188-594-7038
quickly regular instructions wake-- carefully unusual braids into the expres
9847.83 Supplier#000008097 RUSSIA
130557.00 Manufacturer#2
xMe97bpE69NzdwLoX 32-375-640-3593
slyly regular dependencies sleep slyly furiously express dep
9847.57 Supplier#000006345 FRANCE
86344.00 Manufacturer#1
VSt3rzk3qG698u6ld8HhOByvrTcSTSvQlDQDag 16-886-766-7945
silent pinto beans should have to snooze carefully along the final reques
9847.57 Supplier#000006345 FRANCE
173827.00 Manufacturer#2
VSt3rzk3qG698u6ld8HhOByvrTcSTSvQlDQDag 16-886-766-7945
silent pinto beans should have to snooze carefully along the final reques
9836.93 Supplier#000007342 RUSSIA
4841.00 Manufacturer#4
JOlK7C1,7xrEZSSOw 32-399-414-5385
final accounts haggle. bold accounts are furiously dugouts. furiously silent asymptotes are slyly
9817.10 Supplier#000002352 RUSSIA
124815.00 Manufacturer#2
4LfoHUZjgjEbAKw TgdKcgOc4D4uCYw 32-551-831-1437
blithely pending packages across the ironic accounts grow slyly after the furiously
9817.10 Supplier#000002352 RUSSIA
152351.00 Manufacturer#3
4LfoHUZjgjEbAKw TgdKcgOc4D4uCYw 32-551-831-1437
blithely pending packages across the ironic accounts grow slyly after the furiously
9739.86 Supplier#000003384 FRANCE
138357.00 Manufacturer#2
o,Z3v4POifevE k9U1b 6J1ucX,I 16-494-913-5925
slyly ironic theodolites hag
9721.95 Supplier#000008757 UNITED KINGDOM
156241.00 Manufacturer#3
Atg6GnM4dT2 33-821-407-2995
ironic, even dolphins above the furiously ironic foxes sleep slyly around the caref
9681.33 Supplier#000008406 RUSSIA
78405.00 Manufacturer#1
,qUuXcftUl 32-139-873-8571
furiously even deposits affix thinly special theodolites. furiou
9643.55 Supplier#000005148 ROMANIA
107617.00 Manufacturer#1
kT4ciVFslx9z4s79p Js825 29-252-617-4850
doggedly even ideas boost furiously against the furiously express
9624.82 Supplier#000001816 FRANCE
34306.00 Manufacturer#3
e7vab91vLJPW xxZnewmnDBpDmxYHrb 16-392-237-6726
blithely regular accounts cajole furiously. regular
9624.78 Supplier#000009658 ROMANIA
189657.00 Manufact urer#1
oE9uBgEfSS4opIcepXyAYM,x 29-748-876-2014
regular deposits haggle. furiously express asympto
9612.94 Supplier#000003228 ROMANIA
120715.00 Manufacturer#2
KDdpNKN3cWu7ZSrbdqp7AfSLxx,qWB 29-325-784-8187
carefully pending accounts serve. furiously close deposits boost slyly. q
9612.94 Supplier#000003228 ROMANIA
198189.00 Manufacturer#4
KDdpNKN3cWu7ZSrbdqp7AfSLxx,qWB 29-325-784-8187
carefully pending accounts serve. furiously close deposits boost slyly. q
9571.83 Supplier#000004305 ROMANIA
179270.00 Manufacturer#2
qNHZ7WmCzygwMPRDO9Ps 29-973-481-1831
furiously final deposits
9558.10 Supplier#000003532 UNITED KINGDOM
88515.00 Manufacturer#4
..... delete lines .....
33094.00 Manufacturer#5
kkYvL6IuvojJgTNG IKkaXQDYgx8ILohj 17-627-663-8014
quickly regular requests are furiously. pending deposits wake
7937.93 Supplier#000009012 ROMANIA
83995.00 Manufacturer#2
iUiTziH,Ek3i4lwSgunXMgrcTzwdb 29-250-925-9690
blithely bold ideas haggle quickly final, regular request
7914.45 Supplier#000001013 RUSSIA
125988.00 Manufacturer#2
riRcntps4KEDtYScjpMIWeYF6mNnR 32-194-698-3365
final, ironic theodolites alongside of the ironic
7912.91 Supplier#000004211 GERMANY
159180.00 Manufacturer#5
2wQRVovHrm3,v03IKzfTd,1PYsFXQFFOG 17-266-947-7315
final requests integrate slyly above the silent, even
7912.91 Supplier#000004211 GERMANY
184210.00 Manufacturer#4
2wQRVovHrm3,v03IKzfTd,1PYsFXQFFOG 17-266-947-7315
final requests integrate slyly above the silent, even
7894.56 Supplier#000007981 GERMANY
85472.00 Manufacturer#4
NSJ96vMROAbeXP 17-963-404-3760
regular, even theodolites integrate carefully. bold, special theodolites are slyly fluffily iron
7887.08 Supplier#000009792 GERMANY
164759.00 Manufacturer#3
Y28ITVeYriT3kIGdV2K8fSZ V2UqT5H1Otz 17-988-938-4296
pending, ironic packages sleep among the carefully ironic accounts. quickly final accounts
7871.50 Supplier#000007206 RUSSIA
104695.00 Manufacturer#1
3w fNCnrVmvJjE95sgWZzvW 32-432-452-7731
furiously dogged pinto beans cajole. bold, express notornis until the slyly pending
7852.45 Supplier#000005864 RUSSIA
8363.00 Manufacturer#4
WCNfBPZeSXh3h,c 32-454-883-3821
blithely regular deposits
7850.66 Supplier#000001518 UNITED KINGDOM
86501.00 Manufacturer#1
ONda3YJiHKJOC 33-730-383-3892
furiously final accounts wake carefully idle requests. even dolphins wake acc
7843.52 Supplier#000006683 FRANCE
11680.00 Manufacturer#4
2Z0JGkiv01Y00oCFwUGfviIbhzCdy 16-464-517-8943
carefully bold accounts doub

100 rows processed.
Query Processed in 9.89 seconds.

```
-- @(#)3.sql 2.1.6.2
-- TPC-H/TPC-R Shipping Priority Query (Q3)
-- Functional Query Definition
-- Approved February 1998


select * from (
select
l_orderkey,
sum(l_extendedprice * (1 - l_discount)) as revenue,
o_orderdate,
o_shippriority
from
customer,
orders,
```

```
lineitem
where
c_mktsegment = 'BUILDING'
and c_custkey = o_custkey
and l_orderkey = o_orderkey
and o_orderdate < to_date( '1995-03-15', 'YYYY-MM-DD')
and l_shipdate > to_date( '1995-03-15', 'YYYY-MM-DD')
group by
l_orderkey,
o_orderdate,
o_shippriority
order by
revenue desc,
o_orderdate)
where rownum <= 10

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


10 rows processed.
Query Processed in 4.48 seconds.


-- @(#)4.sql              2.1.6.2
-- TPC-H/TPC-R Order Priority Checking Query (Q4)
-- Functional Query Definition
-- Approved February 1998


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

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

5 rows processed.
Query Processed in 4.26 seconds.


-- @(#)5.sql              2.1.6.2
-- TPC-H/TPC-R Local Supplier Volume Query (Q5)
-- Functional Query Definition
-- Approved February 1998


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

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


5 rows processed.
Query Processed in 5.89 seconds.


-- @(#)6.sql              2.1.6.2
-- TPC-H/TPC-R Forecasting Revenue Change Query (Q6)
-- Functional Query Definition
-- Approved February 1998


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

REVENUE
123141078.23


1 row processed.
Query Processed in 2.66 seconds.
```

```
-- @(#)7.sql    2.1.6.2
-- TPC-H/TPC-R Volume Shipping Query (Q7)
-- Functional Query Definition
-- Approved February 1998


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

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


4 rows processed.
Query Processed in 8.20 seconds.


-- @(#)8a.sql           2.1.6.2
-- TPC-H/TPC-R National Market Share Query (Q8)
-- Variant A
-- Approved February 1998


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

O_YEAR               MKT_SHARE
1995.00              0.04
1996.00              0.04


2 rows processed.
Query Processed in 9.20 seconds.


-- @(#)9.sql            2.1.6.2
-- TPC-H/TPC-R Product Type Profit Measure Query (Q9)
-- Functional Query Definition
-- Approved February 1998


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

```
                                nation
where
s_suppkey = l_suppkey
and ps_suppkey = l_suppkey
and ps_partkey = l_partkey
and p_partkey = l_partkey
and o_orderkey = l_orderkey
and s_nationkey = n_nationkey
and p_name like '%green%'
) profit
group by
nation,
o_year
order by
nation,
o_year desc
```

NATION	O_YEAR	SUM_PROFIT
ALGERIA	1998.00	31342867.23
ALGERIA	1997.00	57138193.02
ALGERIA	1996.00	56140140.13
ALGERIA	1995.00	53051469.65
ALGERIA	1994.00	53867582.13
ALGERIA	1993.00	54942718.13
ALGERIA	1992.00	54628034.71
ARGENTINA	1998.00	30211185.71
ARGENTINA	1997.00	50805741.75
ARGENTINA	1996.00	51923746.58
ARGENTINA	1995.00	49298625.77
ARGENTINA	1994.00	50835610.11
ARGENTINA	1993.00	51646079.18
ARGENTINA	1992.00	50410314.99
BRAZIL	1998.00	27217924.38
BRAZIL	1997.00	48378669.20
BRAZIL	1996.00	50482870.36
BRAZIL	1995.00	47623383.63
BRAZIL	1994.00	47840165.73
BRAZIL	1993.00	49054694.04
BRAZIL	1992.00	48667639.08
CANADA	1998.00	30379833.77
CANADA	1997.00	50465052.31
CANADA	1996.00	52560501.39
CANADA	1995.00	52375332.81
CANADA	1994.00	52600364.66
CANADA	1993.00	52644504.07
CANADA	1992.00	53932871.70
CHINA	1998.00	31075466.16
CHINA	1997.00	50551874.45
CHINA	1996.00	51039293.88
CHINA	1995.00	49287534.62
CHINA	1994.00	50851090.07
CHINA	1993.00	54229629.83
CHINA	1992.00	52400529.37
EGYPT	1998.00	29054433.39
EGYPT	1997.00	50627611.45
EGYPT	1996.00	49542212.84
EGYPT	1995.00	48311550.32
EGYPT	1994.00	49790644.74
EGYPT	1993.00	48904292.97
EGYPT	1992.00	49434932.62
ETHIOPIA	1998.00	28040717.27
ETHIOPIA	1997.00	47455009.87
ETHIOPIA	1996.00	46491097.57
ETHIOPIA	1995.00	46804449.30
ETHIOPIA	1994.00	48516143.92
ETHIOPIA	1993.00	46551891.56
ETHIOPIA	1992.00	44934648.64
FRANCE	1998.00	32226407.84
FRANCE	1997.00	47121485.86
FRANCE	1996.00	47263135.50
FRANCE	1995.00	47275997.57
FRANCE	1994.00	47067209.33
FRANCE	1993.00	51163370.11
FRANCE	1992.00	47846235.33
GERMANY	1998.00	28624942.66
GERMANY	1997.00	49309074.88
GERMANY	1996.00	49918683.17
GERMANY	1995.00	52650718.72
GERMANY	1994.00	50346900.42
GERMANY	1993.00	50991895.81
GERMANY	1992.00	48274126.10
INDIA	1998.00	29943144.35
INDIA	1997.00	50665453.23
INDIA	1996.00	50283092.29
INDIA	1995.00	50006774.64
INDIA	1994.00	48995190.76
INDIA	1993.00	50286902.85
INDIA	1992.00	50850329.40
INDONESIA	1998.00	27672340.00
INDONESIA	1997.00	50512145.73
INDONESIA	1996.00	51653060.12
INDONESIA	1995.00	51508779.59
INDONESIA	1994.00	52817950.32
INDONESIA	1993.00	47959994.96
INDONESIA	1992.00	51776605.03
IRAN	1998.00	29065736.24
IRAN	1997.00	50042063.05
IRAN	1996.00	50926653.19
IRAN	1995.00	51249667.65
IRAN	1994.00	50337085.87
IRAN	1993.00	51730763.49
IRAN	1992.00	49955856.56
IRAQ	1998.00	31624551.00
IRAQ	1997.00	55121749.02
IRAQ	1996.00	55897663.79
IRAQ	1995.00	54815472.52
IRAQ	1994.00	54408516.13
IRAQ	1993.00	53633167.98
IRAQ	1992.00	55891939.34
JAPAN	1998.00	27934179.67
JAPAN	1997.00	44517162.55
JAPAN	1996.00	42545606.12
JAPAN	1995.00	43749356.40
JAPAN	1994.00	44840243.07
JAPAN	1993.00	44660015.53
JAPAN	1992.00	45410249.12
JORDAN	1998.00	26901488.58
JORDAN	1997.00	45471878.41
JORDAN	1996.00	46794325.79
JORDAN	1995.00	45178828.58
JORDAN	1994.00	45333636.51
JORDAN	1993.00	47971496.10
JORDAN	1992.00	44717239.18
KENYA	1998.00	28597614.34
KENYA	1997.00	47949733.73
KENYA	1996.00	46886924.62
KENYA	1995.00	46072338.76
KENYA	1994.00	45772061.17
KENYA	1993.00	46308728.23
..... delete lines		
PERU	1992.00	47523899.05
ROMANIA	1998.00	30368667.40
ROMANIA	1997.00	50365683.85
ROMANIA	1996.00	49598999.01
ROMANIA	1995.00	47537642.87
ROMANIA	1994.00	51455283.01
ROMANIA	1993.00	50407136.89
ROMANIA	1992.00	48185385.13
RUSSIA	1998.00	28322384.03
RUSSIA	1997.00	50106685.18
RUSSIA	1996.00	51753342.43
RUSSIA	1995.00	49215820.36

```
RUSSIA                  1994.00         52205666.44
RUSSIA                  1993.00         51860230.03
RUSSIA                  1992.00         53251677.15
SAUDI ARABIA            1998.00         31541259.81
SAUDI ARABIA            1997.00         52438750.81
SAUDI ARABIA            1996.00         52543737.82
SAUDI ARABIA            1995.00         52938696.53
SAUDI ARABIA            1994.00         51389601.97
SAUDI ARABIA            1993.00         52937508.88
SAUDI ARABIA            1992.00         54843459.64
UNITED KINGDOM          1998.00         28494874.00
UNITED KINGDOM          1997.00         49381810.90
UNITED KINGDOM          1996.00         51386853.96
UNITED KINGDOM          1995.00         51509586.79
UNITED KINGDOM          1994.00         48086499.71
UNITED KINGDOM          1993.00         49166827.22
UNITED KINGDOM          1992.00         49349122.08
UNITED STATES           1998.00         25126238.95
UNITED STATES           1997.00         50077306.42
UNITED STATES           1996.00         48048649.47
UNITED STATES           1995.00         48809032.42
UNITED STATES           1994.00         49296747.18
UNITED STATES           1993.00         48029946.80
UNITED STATES           1992.00         48671944.50
VIETNAM                 1998.00         30442736.06
VIETNAM                 1997.00         50309179.79
VIETNAM                 1996.00         50488161.41
VIETNAM                 1995.00         49658284.61
VIETNAM                 1994.00         50596057.26
VIETNAM                 1993.00         50953919.15
VIET NAM                1992.00         49613838.32


175 rows processed.
Query Processed in 3.74 seconds.


-- @(#)10.sql           2.1.6.2
-- TPC-H/TPC-R Returned Item Reporting Query (Q10)
-- Functional Query Definition
-- Approved February 1998


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

C_CUSTKEY           C_NAME                  REVENUE
C_ACCTBAL           N_NAME
C_ADDRESS                       C_PHONE
C_COMMENT
57040.00            Customer#000057040      734235.25
632.87              JAPAN
Eioyzjf4pp                      22-895-641-3466
requests sleep blithely about the furiously i
143347.00           Customer#000143347      721002.69
2557.47             EGYPT
1aReFYv,Kw4                     14-742-935-3718
fluffily bold excuses haggle finally after the u
60838.00            Customer#000060838      679127.31
2454.77             BRAZIL
64EaJ5vMAHWJlBOxJklpNc2RJiWE            12-913-494-9813
furiously even pinto beans integrate under the ruthless foxes; ironic,
even dolphins across the slyl
101998.00           Customer#000101998      637029.57
3790.89             UNITED KINGDOM
01c9CILnNtfOQYmZj               33-593-865-6378
accounts doze blithely! enticing, final deposits sleep blithely special
accounts. slyly express accounts pla
125341.00           Customer#000125341      633508.09
4983.51             GERMANY
S29ODD6bceU8QSuuEJznkNaK                17-582-695-5962
quickly express requests wake quickly blithely
25501.00            Customer#000025501      620269.78
7725.04             ETHIOPIA
 W556MXuoiaYCCZamJI,Rn0B4ACUGdkQ8DZ     15-874-808-6793
quickly special requests sleep evenly among the special deposits. special
deposi
115831.00           Customer#000115831      596423.87
5098.10             FRANCE
rFeBbEEyk dl ne7zV5fDrmiq1oK09wV7pxqCgIc 16-715-386-3788
carefully bold excuses sleep alongside of the thinly idle
84223.00            Customer#000084223      594998.02
528.65              UNITED KINGDOM
nAVZCs6BaWap rrM27N 2qBnzc5WBauxbA      33-442-824-8191
pending, final ideas haggle final requests. unusual, regular asymptotes
affix according to the even foxes.
54289.00            Customer#000054289      585603.39
5583.02             IRAN
vXCxoCsU0Bad5JQI ,oobkZ                 20-834-292-4707
express requests sublate blithely regular requests. regular, even ideas
solve.
39922.00            Customer#000039922      584878.11
7321.11             GERMANY
Zgy4s50l2GKN4pLDPBU8m342gIw6R           17-147-757-8036
even pinto beans haggle. slyly bold accounts inte
6226.00             Customer#000006226      576783.76
2230.09             UNITED KINGDOM
8gPu8,NPGkfyQQ0hcIYUGPIBWc,ybP5g,       33-657-701-3391
quickly final requests against the regular instructions wake blithely final
instructions. pa
922.00              Customer#000000922      576767.53
3869.25             GERMANY
Az9RFaut7NkPnc5zSD2PwHgVwr4jRzq         17-945-916-9648
boldly final requests cajole blith
147946.00           Customer#000147946      576455.13
2030.13             ALGERIA
iANyZHjqhyy7Ajah0pTrYyhJ                10-886-956-3143
furiously even accounts are blithely above the furiousl
115640.00           Customer#000115640      569341.19
6436.10             ARGENTINA
```

```
Vtgfia9qI 7EpHgecU1X 11-411-543-4901
final instructions are slyly according to the
73606.00 Customer#000073606 568656.86
1785.67 JAPAN
xuR0Tro5yChDfOCrjkd2ol 22-437-653-6966
furiously bold orbits about the furiously busy requests wake across the
furiously quiet theodolites. d
110246.00 Customer#000110246 566842.98
7763.35 VIETNAM
7KzflgX MDOq7sOkI 31-943-426-9837
dolphins sleep blithely among the slyly final
142549.00 Customer#000142549 563537.24
5085.99 INDONESIA
ChqEoK43OysjdHbtKCp6dKqjNyvvi9 19-955-562-2398
regular, unusual dependencies boost slyly; ironic attainments nag fluffily
into the unusual packages?
146149.00 Customer#000146149 557254.99
1791.55 ROMANIA
s87fvzFQpU 29-744-164-6487
silent, unusual requests detect quickly slyly regul
52528.00 Customer#000052528 556397.35
551.79 ARGENTINA
NFztyTOR10UOJ 11-208-192-3205
unusual requests detect. slyly dogged theodolites use slyly. deposit
23431.00 Customer#000023431 554269.54
3381.86 ROMANIA
HgiV0phqhaIa9aydNoIlb 29-915-458-2654
instructions nag quickly. furiously bold accounts cajol


20 rows processed.
Query Processed in 0.94 seconds.


-- @(#)11.sql 2.1.6.2
-- TPC-H/TPC-R Important Stock Identification Query (Q11)
-- Functional Query Definition
-- Approved February 1998


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

PS_PARTKEY VALUE
129760.00 17538456.86
166726.00 16503353.92
191287.00 16474801.97
161758.00 16101755.54
34452.00 15983844.72
139035.00 15907078.34
9403.00 15451755.62
154358.00 15212937.88
38823.00 15064802.86
85606.00 15053957.15
33354.00 14408297.40
154747.00 14407580.68
82865.00 14235489.78
76094.00 14094247.04
222.00 13937777.74
121271.00 13908336.00
55221.00 13716120.47
22819.00 13666434.28
76281.00 13646853.68
85298.00 13581154.93
85158.00 13554904.00
139684.00 13535538.72
31034.00 13498025.25
87305.00 13482847.04
10181.00 13445148.75
62323.00 13411824.30
26489.00 13377256.38
96493.00 13339057.83
56548.00 13329014.97
55576.00 13306843.35
159751.00 13306614.48
92406.00 13287414.50
182636.00 13223726.74
199969.00 13135288.21
62865.00 13001926.94
7284.00 12945298.19
197867.00 12944510.52
11562.00 12931575.51
75165.00 12916918.12
97175.00 12911283.50
140840.00 12896562.23
65241.00 12890600.46
166120.00 12876927.22
9035.00 12863828.70
144616.00 12853549.30
176723.00 12832309.74
170884.00 12792136.58
29790.00 12723300.33
95213.00 12555483.73
183873.00 12550533.05
171235.00 12476538.30
21533.00 12437821.32
17290.00 12432159.50
156397.00 12260623.50
122611.00 12222812.98
139155.00 12220319.25
146316.00 12215800.61
171381.00 12199734.52
198633.00 12078226.95
167417.00 12046637.62
59512.00 12043468.76
31688.00 12034893.64
159586.00 12001505.84
8993.00 11963814.30
120302.00 11857707.55
43536.00 11779340.52
9552.00 11776909.16
86223.00 11772205.08
53776.00 11758669.65
131285.00 11616953.74
91628.00 11611114.83
169644.00 11567959.72
182299.00 11567462.05
```

```
33107.00          11453818.76
104184.00         11436657.44
67027.00          11419127.14
176869.00         11371451.71
30885.00          11369674.79
54420.00          11345076.88
72240.00          11313951.05
178708.00         11294635.17
81298.00          11273686.13
158324.00         11243442.72
117095.00         11242535.24
176793.00         11237733.38
86091.00          11177793.79
116033.00         11145434.36
129058.00         11119112.20
193714.00         11104706.39
117195.00         11077217.96
49851.00          11043701.78
19791.00          11030662.62
75800.00          11012401.62
161562.00         10996371.69
10119.00          10980015.75
39185.00          10970042.56
47223.00          10950022.13
175594.00         10942923.05
111295.00         10893675.61
155446.00         10852764.57
156391.00         10839810.38
40884.00          10837234.19
141288.00         10837130.21
152388.00         10830977.82
33449.00          10830858.72
149035.00         10826130.02
162620.00         10814275.68
118324.00         10791788.10
38932.00          10777541.75
121294.00         10764225.22
48721.00          10762582.49
63342.00          10740132.60
5614.00           10724668.80
62266.00          10711143.10
100202.00         10696675.55
197741.00         10688560.72
..... delete lines .....
71518.00          7932261.69
72922.00          7930400.64
146699.00         7929167.40
92387.00          7928972.67
186289.00         7928786.19
95952.00          7927972.78
196514.00         7927180.70
4403.00           7925729.04
2267.00           7925649.37
45924.00          7925047.68
11493.00          7916722.23
104478.00         7916253.60
166794.00         7913842.00
161995.00         7910874.27
23538.00          7909752.06
41093.00          7909579.92
112073.00         7908617.57
92814.00          7908262.50
88919.00          7907992.50
79753.00          7907933.88
108765.00         7905338.98
146530.00         7905336.60
71475.00          7903367.58
36289.00          7901946.50
61739.00          7900794.00
52338.00          7898638.08
194299.00         7898421.24
105235.00         7897829.94
77207.00          7897752.72
96712.00          7897575.27
10157.00          7897046.25
171154.00         7896814.50
79373.00          7896186.00
113808.00         7893353.88
27901.00          7892952.00
128820.00         7892882.72
25891.00          7890511.20
122819.00         7888881.02
154731.00         7888301.33
101674.00         7879324.60
51968.00          7879102.21
72073.00          7877736.11
5182.00           7874521.73

1048 rows processed.
Query Processed in 13.19 seconds.


-- @(#)12.sql          2.1.6.2
-- TPC-H/TPC-R Shipping Modes and Order Priority Query (Q12)
-- Functional Query Definition
-- Approved February 1998


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

L_SHIPMODE HIGH_LINE_COUNT       LOW_LINE_COUNT
MAIL       6202.00               9324.00
SHIP       6200.00               9262.00


2 rows processed.
Query Processed in 1.71 seconds.


-- @(#)13.sql          2.1.6.2
-- TPC-H/TPC-R Customer Distribution Query (Q13)
-- Functional Query Definition
-- Approved February 1998
```

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

```
C_COUNT             CUSTDIST
0.00                50004.00
9.00                6641.00
10.00               6566.00
11.00               6058.00
8.00                5949.00
12.00               5553.00
13.00               4989.00
19.00               4748.00
7.00                4707.00
18.00               4625.00
15.00               4552.00
17.00               4530.00
14.00               4484.00
20.00               4461.00
16.00               4323.00
21.00               4217.00
22.00               3730.00
6.00                3334.00
23.00               3129.00
24.00               2622.00
25.00               2079.00
5.00                1972.00
26.00               1593.00
27.00               1185.00
4.00                1033.00
28.00               869.00
29.00               559.00
3.00                398.00
30.00               373.00
31.00               235.00
2.00                144.00
32.00               128.00
33.00               71.00
34.00               48.00
35.00               33.00
1.00                23.00
36.00               17.00
37.00               7.00
40.00               4.00
38.00               4.00
39.00               2.00
41.00               1.00
```

```
42 rows processed.
Query Processed in 3.06 seconds.
```

```
-- @(#)14.sql          2.1.6.2
-- TPC-H/TPC-R Promotion Effect Query (Q14)
-- Functional Query Definition
-- Approved February 1998


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

```
PROMO_REVENUE
16.38
```

```
1 row processed.
Query Processed in 2.70 seconds.
```

```
-- @(#)15.sql   2.1.6.2
-- TPC-H/TPC-R Top Supplier Query (Q15a)
-- Functional Query Definition
-- Approved February 1998


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

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

1 row processed.
Query Processed in 10.16 seconds.

```
-- @(#)16.sql           2.1.6.2
-- TPC-H/TPC-R Parts/Supplier Relationship Query (Q16)
-- Functional Query Definition
-- Approved February 1998


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

18314 rows processed.
Query Processed in 2.49 seconds.
```

```
-- @(#)17.sql            2.1.6.2
-- TPC-H/TPC-R Small-Quantity-Order Revenue Query (Q17)
-- Functional Query Definition
-- Approved February 1998
```

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

AVG_YEARLY
348406.05


1 row processed.
Query Processed in 2.93 seconds.
```

```
-- @(#)18.sql            2.1.6.2
-- TPC-H/TPC-R Large Volume Customer Query (Q18)
-- Function Query Definition
-- Approved February 1998


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

C_NAME                C_CUSTKEY            O_ORDERKEY
O_ORDERDATE
O_TOTALPRICE          SUM(L_QUANTITY)
```

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

```
Customer#000112987      112987.00       4439686.00      1996-
09-17
418161.49       305.00
Customer#000012599      12599.00        4259524.00      1998-
02-12
415200.61       304.00
Customer#000105410      105410.00       4478371.00      1996-
03-05
412754.51       302.00
Customer#000149842      149842.00       5156581.00      1994-
05-30
411329.35       302.00
Customer#000010129      10129.00        5849444.00      1994-
03-21
409129.85       309.00
Customer#000069904      69904.00        1742403.00      1996-
10-19
408513.00       305.00
Customer#000017746      17746.00        6882.00         1997-
04-09
408446.93       303.00
Customer#000013072      13072.00        1481925.00      1998-
03-15
399195.47       301.00
Customer#000082441      82441.00        857959.00       1994-
02-07
382579.74       305.00
Customer#000088703      88703.00        2995076.00      1994-
01-30
363812.12       302.00


57 rows processed.
Query Processed in 3.50 seconds.


-- @(#)19.sql           2.1.6.2
-- TPC-H/TPC-R Discounted Revenue Query (Q19)
-- Functional Query Definition
-- Approved February 1998


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

REVENUE
3083843.06


1 row processed.
Query Processed in 2.62 seconds.


-- @(#)20.sql           2.1.6.2
-- TPC-H/TPC-R Potential Part Promotion Query (Q20)
-- Function Query Definition
-- Approved February 1998


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 7hMlCof1Y5zLFg
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
Supplier#000002020 jB6r1d7MxP6co
Supplier#000002022 dwebGX7Id2pc25YvY33
Supplier#000002036 20ytTtVObjKUUI2WCB0A
Supplier#000002204 uYmlr46C06udCqanj0KiRsoTQakZsEyssL
Supplier#000002243 nSOEV3JeOU79
Supplier#000002245 hz2qWXW VjOyKhqPYMoEwz6zFkrTaDM
Supplier#000002282 ES21K9dxoW1I1TzWCj7ekdlNwSWnv1Z
6mQ,BKn
Supplier#000002303 nCoWfpB6YOymbgOht7ltfklpkHl
Supplier#000002373 RzHSxOTQmElCjxIBiVA52Z JB58rJhPRylR
Supplier#000002419 qydBQd14I5l5mVXa4fYY
Supplier#000002481 nLKHUOn2Ml9TOA06Znq9GEMcIlMO2
Supplier#000002571 JZUugz04c iJFLrlGsz9O N,W 1rVHNIReyq
Supplier#000002585 CsPoKpw2QuTY4AV1NkWuttneIa4SN
Supplier#000002630 ZIQAvjNUY9KH5ive zm7k VlPiDl7CCo21
Supplier#000002719 4nnzQI2CbqREQUuIsXTBVUkaP4mNS3
Supplier#000002721 HVdFAN2JHMQSpKm
Supplier#000002730 lIFxR4fzm31C6,muzJwl84z
Supplier#000002775 yDclaDaBD4ihH
Supplier#000002853 rTNAOItXka
Supplier#000002875 6JgMi
9Qt6VmwL3Ltt1SRlKww0keLQ,RAZa
Supplier#000002934 m,trBENywSA rwg3DhB
Supplier#000002941 Naddba 8YTEKekZyP0
Supplier#000002960 KCPCEsRGGo6vx8TygHh60nAYf9rStQT2T
Supplier#000002980 B9k9yVsyaXvWktOSHezqHiAEp9id0SKzkw
Supplier#000003062 LSQNgqY1xnOzz9zBCapy7HwOZQ
Supplier#000003087 ANwe8QsZ4rgj1HSqVz991eWQ
Supplier#000003089 s5b VCIZqMSZVa r g7LTdcg29GbTE7rI1x
Supplier#000003095 HxON3jJhUi3zjt,r mTD
Supplier#000003201 E87yws6I,t0qNs4QW7UzExKiJnJDZWue
Supplier#000003213 pxrRP4irQ1VoyfQ,dTf3
Supplier#000003241 j06SU,LS9O3mwjAMOViANeIhb
Supplier#000003275 9xO4nyJ2QJcX6vGf
Supplier#000003288 EDdfNt7E5Uc,xLTupoIgYL4yY7ujh,
Supplier#000003313
El2I7we,049SPrvomUm4hZwJoOhZkvLxLJXgVH
Supplier#000003314
jnisU8MzqO4iUB3zsPcrysMw3DDUojS4q7LD
Supplier#000003380 jPv0V,pszouuFT3YsAqlP,kxT3u,gTFiEbRt,x
Supplier#000003403 e3X2o ,KCG9tsHji8A XXCxiF2hZWBw
Supplier#000003421 Sh3dt9W5oeofFWovnFhrg,
Supplier#000003441 zvFJIzS,oUuShHjpcX
Supplier#000003590 sy79CMLxqb,Cbo
Supplier#000003607 lNqFHQYjwSAkf
Supplier#000003625
qY588W0Yk5iaUy1RXTgNrEKrMAjBYHcKs
Supplier#000003656 eEYmmO2gmD JdfG32XtDgJV,db56
Supplier#000003782 iVsPZg7bk06TqNMwi0LKbLUrC1zmrg
Supplier#000003918 meRvRCsJoAbfqd0Re4
Supplier#000003941 Pmb05mQfBMS618O7WKqZJ 9vyv
Supplier#000003994 W00LZp3NjK0
Supplier#000004005 V723F1wCy2eA4OgIu8TjBtOVUHp
Supplier#000004033 ncsAhv9Je,kFXTNjfb2
Supplier#000004140 0hL7DJyYjcHL
Supplier#000004165 wTJ2dZNQA8P2oi99N6DT47ndHy,XKD2
Supplier#000004207 tF64pwiOM4IkWjN3mS,e06WuAjLx
Supplier#000004236 dl,HPtJmGipxYsSqn9wmqkuWjst,mCeJ8O6T
Supplier#000004246 Xha aXQF7u4qU3LsHD
Supplier#000004278 bBddbpBxIVp Di9
Supplier#000004343 GK3sbopqrQEkWLMvVBFCG
Supplier#000004346 S3076LEOwo
Supplier#000004388 VfZ l1J,mwp4aS
Supplier#000004406
Ah0ZaLu6VwufPWUz,7kbXgYZhauEaHqGIg
Supplier#000004430 yvSsKNSTL5HLXBET4luOsPNLxKzAMk
Supplier#000004522 xXtCKwsZDArxIBGDfzX2PgobGZsBg
Supplier#000004527 p pVXCnxgcklWF6A1o3OHY3qW6
Supplier#000004542 NJSbLJDroYG2y1r3rDiKg
Supplier#000004574 1HvGwnVueZ5CIndc
Supplier#000004655 67NqBc4 t3PG3F8aO IsqWNq4kGaPowYL
Supplier#000004701 6jX4u47URzIMHf
Supplier#000004711 bEzjp1QdQu ls2ERMxv0km vn6bu2zXlL1
Supplier#000004987 UFx1upJ8MvOvgFjA8
Supplier#000005000 DeX804 w0H8FrCUvahgy ilbuzBX3NK
Supplier#000005100 OfvYPs3Io,wEvvLHNaLuCX
Supplier#000005192 JDp4rhXiDw0kf6RH
Supplier#000005195 Woi3b2ZaicPh ZSfu1EfXhE
..... delete lines .....
Supplier#000007885 u3sicchh5ZpyTUpN1cJKNcAoabIWgY
Supplier#000007918 r,v9mBQ6LoEYyj1
Supplier#000007926 ErzCF80K9Uy
Supplier#000007957 ELwnio14ssoU1 dRyZIL OK3Vtzb
Supplier#000007965 F7Un5lJ7p5hhj
Supplier#000007968
DsF9UlZ2Fo6HXN9aErvyg1ikHoD582HSGZpP
Supplier#000007998 LnASFBfYRFOo9d6d,asBvVq9Lo2P
Supplier#000008168 aOa82a8ZbKCnfDLX
Supplier#000008231 IK7eGw Yj90sTdpsP,vcqWxLB
Supplier#000008243 2AyePMkDqmzVzjGTizXthFLo8h
EiudCMxOmIIG
Supplier#000008275 BlbNDfWg,gpXKQlLN
Supplier#000008323 75I18sZmASwm
POeheRMdj9tmpyeQ,Bf CXN5BIAb
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 Processed in 3.09 seconds.


-- @(#)21.sql           2.1.6.2
-- TPC-H/TPC-R Suppliers Who Kept Orders Waiting Query (Q21)
-- Functional Query Definition
-- Approved February 1998


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

S_NAME                NUMWAIT
Supplier#000002829      20.00
Supplier#000005808      18.00
Supplier#000000262      17.00
Supplier#000000496      17.00
Supplier#000002160      17.00
Supplier#000002301      17.00
Supplier#000002540      17.00
Supplier#000003063      17.00
Supplier#000005178      17.00
Supplier#000008331      17.00
Supplier#000002005      16.00
Supplier#000002095      16.00
Supplier#000005799      16.00
Supplier#000005842      16.00
Supplier#000006450      16.00
Supplier#000006939      16.00
Supplier#000009200      16.00
Supplier#000009727      16.00
Supplier#000000486      15.00
Supplier#000000565      15.00
Supplier#000001046      15.00
Supplier#000001047      15.00
Supplier#000001161      15.00
Supplier#000001336      15.00
Supplier#000001435      15.00
Supplier#000003075      15.00
Supplier#000003335      15.00
Supplier#000005649      15.00
Supplier#000006027      15.00
Supplier#000006795      15.00
Supplier#000006800      15.00
Supplier#000006824      15.00
Supplier#000007131      15.00
Supplier#000007382      15.00
Supplier#000008913      15.00
Supplier#000009787      15.00
Supplier#000000633      14.00
Supplier#000001960      14.00
Supplier#000002323      14.00
Supplier#000002490      14.00
Supplier#000002993      14.00
Supplier#000003101      14.00
Supplier#000004489      14.00
Supplier#000005435      14.00
Supplier#000005583      14.00
Supplier#000005774      14.00
Supplier#000007579      14.00
Supplier#000008180      14.00
Supplier#000008695      14.00
Supplier#000009224      14.00
Supplier#000000357      13.00
Supplier#000000436      13.00
Supplier#000000610      13.00
Supplier#000000788      13.00
Supplier#000000889      13.00
Supplier#000001062      13.00
Supplier#000001498      13.00
Supplier#000002056      13.00
Supplier#000002312      13.00
Supplier#000002344      13.00
Supplier#000002596      13.00
Supplier#000002615      13.00
Supplier#000002978      13.00
Supplier#000003048      13.00
Supplier#000003234      13.00
Supplier#000003727      13.00
Supplier#000003806      13.00
Supplier#000004472      13.00
```

```
Supplier#000005236      13.00
Supplier#000005906      13.00
Supplier#000006241      13.00
Supplier#000006326      13.00
Supplier#000006384      13.00
Supplier#000006394      13.00
Supplier#000006624      13.00
Supplier#000006629      13.00
Supplier#000006682      13.00
Supplier#000006737      13.00
Supplier#000006825      13.00
Supplier#000007021      13.00
Supplier#000007417      13.00
Supplier#000007497      13.00
Supplier#000007602      13.00
Supplier#000008134      13.00
Supplier#000008234      13.00
Supplier#000009435      13.00
Supplier#000009436      13.00
Supplier#000009564      13.00
Supplier#000009896      13.00
Supplier#000000379      12.00
Supplier#000000673      12.00
Supplier#000000762      12.00
Supplier#000000811      12.00
Supplier#000000821      12.00
Supplier#000001337      12.00
Supplier#000001916      12.00
Supplier#000001925      12.00
Supplier#000002039      12.00
Supplier#000002357      12.00
Supplier#000002483      12.00

100 rows processed.
Query Processed in 7.92 seconds.
```

```
-- @(#)22.sql   2.1.4.2
-- TPC-H/TPC-R Global Sales Opportunity Query (Q22)
-- Functional Query Definition
-- Approved February 1998

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_acct bal > 0.00
and substr(c_phone, 1, 2) in
('13', '31', '23', '29', '30', '18', '17')
)
and not exists (
select
*
from
orders
where
o_custkey = c_custkey
)
) custsale
group by
cntrycode
order by
cntrycode

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

7 rows processed.
Query Processed in 7.22 seconds.
```

Appendix E Seed and Input Parameters

E.1 seed

0917083008

E.2 qp1.0

14 1993-06-01
2 37 TIN EUROPE
9 mint
20 dim 1997-01-01 FRANCE
6 1993-01-01 0.06 24
17 Brand#42 LG BOX
18 313
8 EGYPT MIDDLE EAST LARGE BURNISHED COPPER
21 VIETNAM
13 special packages
3 HOUSEHOLD 1995-03-21
22 13 18 24 11 19 29 10
16 Brand#54 SMALL BRUSHED 15 20 38 8 21 42 18 2
4 1996-03-01
11 RUSSIA 0.0000000100
15 1996-03-01
1 85
10 1994-09-01
19 Brand#22 Brand#52 Brand#22 4 17 30
5 AMERICA 1993-01-01
7 KENYA EGYPT
12 AIR REG AIR 1993-01-01

E.3 qp1.1

21 JORDAN
3 AUTOMOBILE 1995-03-06
18 314
5 ASIA 1993-01-01
11 IRAN 0.0000000100
7 FRANCE VIETNAM
6 1993-01-01 0.03 25
20 papaya 1995-01-01 VIETNAM
17 Brand#44 LG PACK
12 SHIP AIR 1993-01-01
16 Brand#44 ECONOMY ANODIZED 16 24 31 6 14 2 4 5
15 1993-11-01
13 special packages
10 1993-06-01
2 25 COPPER AFRICA
8 VIETNAM ASIA LARGE ANODIZED TIN
14 1993-09-01
19 Brand#24 Brand#35 Brand#21 9 18 26
9 linen
22 31 21 16 20 28 17 32
1 93
4 1993-12-01

E.4 qp1.2

6 1994-01-01 0.09 25
17 Brand#41 LG DRUM
14 1994-01-01
16 Brand#24 STANDARD PLATED 19 25 44 23 18 46 50 42
19 Brand#21 Brand#23 Brand#15 5 19 22
10 1994-03-01
9 lace
2 13 STEEL EUROPE
15 1996-06-01
8 JORDAN MIDDLE EAST MEDIUM POLISHED TIN
5 EUROPE 1994-01-01
22 21 14 24 20 30 16 25
12 FOB AIR 1993-01-01
7 UNITED KINGDOM JORDAN
13 special packages
18 312
1 101
4 1996-07-01
20 blue 1994-01-01 IRAQ
3 FURNITURE 1995-03-23
11 UNITED KINGDOM 0.0000000100
21 ETHIOPIA

E.5 qp1.3

8 ETHIOPIA AFRICA MEDIUM BURNISHED TIN
5 MIDDLE EAST 1994-01-01
4 1994-04-01
6 1994-01-01 0.06 24
17 Brand#43 MED BOX
7 MOROCCO ETHIOPIA
1 109
18 313
22 17 16 28 34 24 19 15
14 1994-04-01
9 grey
10 1994-12-01
15 1994-03-01
11 IRAQ 0.0000000100
20 linen 1997-01-01 ALGERIA
2 1 NICKEL AMERICA
21 UNITED KINGDOM
19 Brand#33 Brand#51 Brand#14 10 20 29
13 special packages
16 Brand#54 MEDIUM POLISHED 19 4 1 50 7 39 28 15
12 MAIL AIR 1994-01-01
3 MACHINERY 1995-03-08

E.6 qp1.4

```
5       AFRICA  1994-01-01
21      MOROCCO
14      1994-07-01
19      Brand#35  Brand#44  Brand#14  5       10
        26
15      1996-10-01
17      Brand#45  MED PACK
12      TRUCK   AIR     1994-01-01
6       1994-01-01        0.04    25
4       1996-11-01
9       forest
8       RUSSIA  EUROPE  SMALL BRUSHED TIN
16      Brand#44  PROMO ANODIZED  24      47      5
        32      30      12      43      26
11      UNITED STATES     0.0000000100
2       39      TIN     EUROPE
10      1993-09-01
18      315
1       117
13      pending   requests
7       GERMANY         RUSSIA
22      14      11      17      21      13
        23      31
3       FURNITURE         1995-03-25
20      tan     1996-01-01        MOROCCO
```

E.7 qp1.5

```
21      GERMANY
15      1994-06-01
4       1994-08-01
6       1994-01-01        0.09    25
7       UNITED STATES     KENYA
16      Brand#25  SMALL BURNISHED         25
        12      13      17      26      8       5
        23
19      Brand#32  Brand#22  Brand#53  10      11
        22
18      312
14      1994-10-01
22      31      22      14      11      21
        20      12
11      JAPAN   0.0000000100
13      pending   requests
3       MACHINERY         1995-03-10
1       64
2       26      COPPER  AMERICA
5       AMERICA         1994-01-01
8       KENYA   AFRICA  SMALL PLATED TIN
20      ghost   1994-01-01        ETHIOPIA
12      RAIL    FOB     1994-01-01
17      Brand#52  MED DRUM
10      1994-07-01
9       dark
```

E.8 qp1.6

```
10      1993-04-01
3       BUILDING          1995-03-27
15      1997-01-01
13      pending   requests
6       1995-01-01        0.07    24
8       FRANCE  EUROPE  SMALL ANODIZED NICKEL
9       chocolate
7       MOZAMBIQUE      FRANCE
4       1997-03-01
11      ALGERIA 0.0000000100
22      34      33      17      22      14
        12      27
18      314
12      REG AIR   RAIL    1995-01-01
1       72
5       ASIA    1995-01-01
16      Brand#55  LARGE POLISHED  8       28
        13      41      47      1       44
        10
2       14      STEEL   MIDDLE EAST
14      1995-02-01
19      Brand#44  Brand#55  Brand#52  6       12
        29
20      red     1997-01-01        SAUDI ARABIA
17      Brand#54  JUMBO BOX
21      UNITED STATES
```

E.9 qp1.7

```
18      315
8       UNITED KINGDOM  EUROPE  STANDARD
POLISHED NICKEL
20      coral   1996-01-01        INDONESIA
21      MOZAMBIQUE
2       2       BRASS   AMERICA
4       1994-11-01
22      23      31      13      19      30
        22      24
17      Brand#55  JUMBO PACK
1       80
11      JORDAN  0.0000000100
9       blush
19      Brand#41  Brand#42  Brand#51  1       13
        25
3       MACHINERY         1995-03-12
13      pending   requests
5       EUROPE  1995-01-01
7       INDIA   UNITED KINGDOM
10      1994-01-01
16      Brand#45  PROMO BRUSHED   16      22
        34      20      43      27      9
        15
6       1995-01-01        0.04    25
14      1995-05-01
15      1994-10-01
12      SHIP    RAIL    1995-01-01
```

E.10 qp1.8

```
19      Brand#43  Brand#25  Brand#45  6       14
        21
1       88
15      1997-05-01
17      Brand#52  JUMBO DRUM
5       MIDDLE EAST       1995-01-01
8       MOROCCO         AFRICA  STANDARD
BURNISHED NICKEL
9       azure
12      FOB     RAIL    1995-01-01
14      1995-08-01
7       ALGERIAMOROCCO
```

4 1997-06-01
3 BUILDING 1995-03-29
20 moccasin 1994-01-01 UNITED STATES
16 Brand#25 MEDIUM BURNISHED 21
19 17 50 47 31
29 3
6 1995-01-01 0.02 25
22 27 23 10 17 15
19 25
10 1994-10-01
13 pending accounts
2 40 TIN MIDDLE EAST
21 INDIA
18 313
11 ARGENTINA 0.0000000100

E.11 qp1.9

8 GERMANY EUROPE PROMO BRUSHED NICKEL
13 pending accounts
2 27 COPPER ASIA
20 antique 1993-01-01 KENYA
17 Brand#54 WRAP BOX
3 HOUSEHOLD 1995-03-14
6 1995-01-01 0.07 24
21 ARGENTINA
18 315
11 KENYA 0.0000000100
19 Brand#51 Brand#13 Brand#45 1 15
29
10 1993-08-01
15 1995-01-01
4 1995-03-01
22 17 27 13 30 32
12 20
1 96
7 PERU GERMANY
12 MAIL RAIL 1995-01-01
9 wheat
14 1995-11-01
5 AFRICA 1995-01-01
16 Brand#15 ECONOMY PLATED 12
17 3 29 30 42 5
24

Appendix F Benchmark Scripts

F.2 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;
```

F.3 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;
```

F.4 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"
```

F.5 gtime.c

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

/*

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

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

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

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

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

   NOTES
     <other useful comments, qualifications, etc.>

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

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

# include <sys/time.h>

main ()
{
  struct timeval tv;

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

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

}

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

F.6 qexecpl.c

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

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

/*

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

   DESCRIPTION
     SQL Execution Engine, Oracle v8,  OCI version

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

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

*/

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

#include "qexecpl.h"

/* Function Prototypes */

extern double gettime();

/* function prototypes from gen.c */

int get_statement();

/* Declare error handling functions */

void sql_error();
```

```
/* Other prototypes */

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

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

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

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

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

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

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

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

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

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

char stmt[SQL_LEN];       /* The SQL statement or comment line.  */
char qn[3];               /* Number of the query being executed  */
char qnp[3];              /* Number of the previous query executed  */
char cmnt[5000];          /* Buffer to save the comment.         */
#ifdef LINUX
FILE *qtemp;      /* fd for query template               */
FILE *logfile;    /* log and report files                */
FILE *rep;
#else
FILE *qtemp = stdin;      /* fd for query template               */
FILE *logfile = stdout;   /* log and report files                */
FILE *rep = stdout;
#endif
void *defbuf;             /* Buffer pointer for ODEFIN           */
int deflen = 0;           /* Size of data type for ODEFIN        */
int deftype = 1;          /* Oracle type number for ODEFIN       */

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

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

/* OCI handles */

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

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

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

void usage() {

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



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

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

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

```
                                        2048,OCI_HTYPE_ERROR);
    else
      (void) OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)msg,
                                        2048,OCI_HTYPE_ENV);
    fprintf(stderr,"%s\n",msg);
    break;
  }

  /* Rollback just in case */

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

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

  SQLexit();

  exit(1);
}

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

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

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

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

  strcpy(logname, argv[1]);

  /* Process  optional parameters  */

  argc -= 1;
  argv += 1;

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

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

  SQLinit();

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

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

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

    switch (retcode) {

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

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

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

      /* Executes the query */
      SQLexec();

      stmt_cnt++;
      qry_cnt++;
      fflush(rep);
      fflush(logfile);
```

```
    /*
    fprintf(logfile,"\nStatement Started at %.2f\n", s_tr_start);
    fprintf(logfile,"Statement Ended at %.2f\n", s_tr_end);

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

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

  /* Get Timing for the last query */

  tr_end = gettime();

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

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

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

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

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

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

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

  fflush(rep);
  fflush(logfile);

  /* Close the query template file */

  fclose(qtemp);

  /*  Disconnect from ORACLE. */

  SQLexit();
  exit(0);
}


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

void SQLinit() {

  int i;

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

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

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

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

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

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

  /* get username and password */

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

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


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

OCIaset(tpcusr,OCI_HTYPE_SESSION,logname,strlen(logname),OCI_
ATTR_USERNAME,
                errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI_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_SESSIO
N,errhp);

/*
```

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

}



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

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

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

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

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

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

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

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

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

  s_tr_start = gettime();

  /* prepare the statement */

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

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

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

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


OCIaget(curq,OCI_HTYPE_STMT,&stmttyp,NULL,OCI_ATTR_STM
T_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_RO
W_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_RO
W_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_RO
W_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;
}


/* memallo c(): 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 = ' ';
}
```

F.7 qexecpl.h

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

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

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

/*
   NAME
     qexecpl.h

   DESCRIPTION
     SQL statement execution front-end header file.

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

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

   EXAMPLES

   NOTES
     <other useful comments, qualifications, etc.>

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

*/

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

#define QSTREAMPL_H

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

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

#include <oratypes.h>

#include <oratypes.h>

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

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

/* some basic definitions */

#define UNAME_LEN 64
#define MAX_FILE_PATH_LEN 128

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

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

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

/* defines and typedefs for query description */

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

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

/* types for describe */

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

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

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

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


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

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

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

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

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

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

extern int errno;

#define SQL_LEN 2048

#ifndef NULL
#define NULL 0
#endif

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

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#ifndef ub1
#define ub1 unsigned char
#endif

#define NA           -1     /* ANSI SQL NULL */
```

```
#define VER7          2
#define NOT_SERIALIZABLE  8177  /* ORA-08177: transaction not
serializable */

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

/* For get_statement */

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


#define OCIhalloc(envh,hndl,htyp) \
   if((status=OCIHandleAlloc((dvoid *)envh,(dvoid
**)hndl,htyp,0,(dvoid **)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_DE
FAULT)) != OCI_SUCCESS) \
      sql_error(errh,status,1); \
   else \
      DISCARD 0


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

#endif /* QSTREAMPL_H */
```

F.8 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}/Ld3dapop
LD4IXCRE=${OUT_DIR}/Ld4ixcre
LD5ANLYZ=${OUT_DIR}/Ld5anlyz


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

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

### on the second node:
remsh titan2 -n "mv /oracle/rdbms/log/alert_tpch2.log
/oracle/rdbms/log/alert_tpch2.log.preAudit.$RUN_ID"
remsh titan2 -n "touch /oracle/rdbms/log/alert_tpch2.log"


echo "Start: load database `date`" >> $SCRIPT_LOG_FILE
dbcre.sh > $LD1DBCRE
sctso.sh > $LD2SCTSO
STIME=`$GTIME`
echo "Start: timed load portion `date`" >> $SCRIPT_LOG_FILE
$FRAME_DIR/bin/tshut >>  $SCRIPT_LOG_FILE
$FRAME_DIR/bin/tstart >>  $SCRIPT_LOG_FILE
dapop.sh > $LD3DAPOP
ixcre.sh > $LD4IXCRE
anl.sh > $LD5ANLYZ
$FRAME_DIR/bin/tshut
$FRAME_DIR/bin/tstart
$KIT_DIR/audit/ckpnt.sh
echo "End: timed load portion `date`" >> $SCRIPT_LOG_FILE

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

echo "Start: dbtables.sql 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

$FRAME_DIR/bin/tshut >>  $SCRIPT_LOG_FILE
$FRAME_DIR/bin/tstart >>  $SCRIPT_LOG_FILE
$KIT_DIR/audit/ckpnt.sh
runTPCHpt ${SCALE_FACTOR} 1 ${RUN_ID}

$FRAME_DIR/bin/tshut >>  $SCRIPT_LOG_FILE
$FRAME_DIR/bin/tstart >>  $SCRIPT_LOG_FILE
$KIT_DIR/audit/ckpnt.sh
runTPCHpt ${SCALE_FACTOR} 2 ${RUN_ID}

sleep 600
# call the auditor: don't tshut >> $SCRIPT_LOG_FILE

cp $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log
$OUT_DIR
rcp titan2:$ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log
$OUT_DIR

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

F.9 runTPCHpt

```
#!/bin/ksh
. $KIT_DIR/env
#set -x
#ECHO=/bin/echo
SCRIPT_DIR=${KIT_DIR}/scripts
UPD_DIR=${KIT_DIR}/update
SRC_DIR=${KIT_DIR}/utils
QRY_DIR=${KIT_DIR}/queries # this is the location of the query
template file
QGEN_DIR=${KIT_DIR}/dbgen
QGEN=${QGEN_DIR}/qgen
QEXEC=${SRC_DIR}

DSS_QUERY=${KIT_DIR}/queries
export DSS_QUERY

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

TPCD_BIN=${KIT_DIR}/audit/bin

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

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

# The defaults

QPROG=${QEXEC}/qexec

usage () {

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

while :
do
  case "$1" in
  -u1) shift; U1PROG=$1;;
  -u2) shift; U2PROG=$1;;
  -p) shift; QPROG=$1;;
# not needed ?   -o) OSTAT=1;;
# not needed ?   -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=${TPCD_RPT}/qtemp.${P ARA}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
```

F.10 runTPCHus

```
E#!/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

rror! Not a valid filename.
```

F.11 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)
#
#
. $KIT_DIR/env
O=${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_1_DOP}

LOGPATH=.
PASSWD=${DATABASE_USER}

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

# perform the update function 1

START=`$GTIME`

# first create the temp tables

sqlplus /NOLOG << !

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

drop directory data_dir;
create directory data_dir as '/flat15/updates';

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

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


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


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


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 ""
```

F.12 runuf2.sh

```
#!/bin/ksh
#
# $Header: runuf2.sh 25-oct-2001.15:56:05 mpoess Exp $
#
# runuf2.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation.  All rights reserved.
#
#    NAME
#      runuf2.sh - <one-line expansion of the name>
#
# DESCRIPTION
#    runuf2.sh [-u <uid/passwd to login>] [-p <program>] <run_id>
#              <scale factor> <pair number> <parallelism>
# USAGE
#   To execute UF2.
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#
. $KIT_DIR/env
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
UTILS_DIR=${KIT_DIR}/utils
GTIME=${UTILS_DIR}/gtime
LOG_DIR=${UPDATE_DIR}/log
PAR_HINT=${UPDATE_2_DOP}
SF=${SCALE_FACTOR}
PASSWD=${DATABASE_USER}

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

SETNUM=$1

i=1
PID=""

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

sqlplus /NOLOG << !

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

drop directory data_dir;
create directory data_dir as '/flat15/updates';

drop table temp_okey_et;
drop table temp_okey;

create table temp_okey_et(
   t_orderkey          number
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
              records delimited by newline
              nobadfile
              nologfile
              fields termin ated by '|'
              missing field values are null
       )
       location (
       'delete.${SETNUM}'))
reject limit unlimited parallel 16;


create table temp_okey (t_orderkey, constraint tokey1 primary
key(t_orderkey))
organization index parallel  16 nologging as select * from
temp_okey_et;
execute dbms_stats.gather_table_stats('tpch' , 'temp_okey',
estimate_percent => 1, degree => 16)


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


delete from (select /*+ use_nl(o) */ o.rowid from orders o, temp_okey 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 ""
```

F.13 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
```

F.14 set_queue

```
#!/sbin/sh

#set -x

for c in \
c1 c13 c27 c41 c3  c15 c29 c43 c5 c17 c31 c45 \
c7 c19 c33 c47 c11 c21 c35 c49 c9 c23 c37 c52 \
c39  c162 c176 c154 c25 c160 c174 c152 c158 c172 c150 c168 \
c156 c170 c164 c166 c54 c66  c78  c90  c56  c68  c80  c92  \
c58 c70 c82 c94 c60 c72 c84 c96 c62 c74 c86 c98 \
c64 c76 c88 c100 c112 c124 c136 c148 c110 c122 c134 c146 \
c108 c120 c132 c144 c106 c118 c130 c142 c104 c116 c128 c140 \
c138 c126 c114 c102
do
  for d in \
  t0d0 t0d1 t0d2 t0d3 t0d4 t0d5 t0d6 t0d7 t1d0 t1d1 t1d2 t1d3 \
  t1d4 t1d5 t1d6 t1d7 t2d0 t2d1 t2d2 t2d3 t2d4 t2d5 t2d6 t2d7 \
  t3d0 t3d1 t3d2 t3d3
  do
    scsictl -m queue_depth=64 /dev/rdsk/${c}${d}
  done
done
```

F.15 tshut (on 1st node)

```
#!/bin/ksh

remsh titan2 -n  /dbms/oracle10g/frame/bin/tshut &

if [ "$1" = "abort" ]; then
sqlplus /NOLOG<< !
connect / as sysdba
shutdown abort
exit
!
else
sqlplus /NOLOG<< ! &
connect / as sysdba
shutdown immediate
exit
!
fi

wait

echo INST 1 DOWN

exit
```

F.16 tshut (on 2nd node)

```
#!/bin/ksh

. ~oracle/.profile

sleep 10

if [ "$1" = "abort" ]; then
sqlplus /NOLOG<< !
connect / as sysdba
shutdown abort
exit
!
else
sqlplus /NOLOG<< !
connect / as sysdba
shutdown immediate
exit
!
echo INST 2 DOWN
fi
```

F.17 tstart (on 1st node)

```
#!/bin/ksh

#. ~oracle/.profile

#set -x
rcp /oracle/dbs/10TB_titan.ora
titan2:/oracle/dbs/
remsh titan2 -n
/dbms/oracle10g/frame/bin/tstart &

mpsched -P RR sqlplus /NOLOG << ! &
connect / as sysdba
startup pfile=/oracle/dbs/10TB_rac1.ora
exit
!
wait
/Lvm/set_queue;

exit
```

F.18 tstart (on 2nd node)

```
#!/bin/ksh

. ~oracle/.profile

#set -x
sleep 10

mpsched -P RR sqlplus /NOLOG << !
connect / as sysdba
startup pfile=/oracle/dbs/10TB_rac2.ora
exit
!
/Lvm/set_queue;
exit
```

```
#!/bin/ksh

. ~oracle/.profile

#set -x
sleep 10

mpsched -P RR sqlplus /NOLOG << !
connect / as sysdba
startup pfile=/oracle/dbs/10TB_rac2.ora
exit
!
```

```
/Lvm/set_queue;
exit
```

Appendix G Price Quotes

The following pages contain the price quotes for the hardware included in this FDR.

Juergen Mueller
HP
Cupertino, CA 95014
December 24, 2004

HP Unix Sales Development
19111 Pruneridge Avenue
Cupertino, CA 95014
(408) 447-2320

HP Integrity Superdome Enterprise Server

TPC-H Rev 2.1.0

Report Date: December 24, 2004

Description	Part Number	Source	Reference Price	Qty	Extended Price	3 yr. Maint. Price
Server Hardware						
Superdome left chassis	A5201A, Opt. 429	1	205,840	2	411,680	
Superdome right chassis	A5202A, Opt. 429	1	218,435	2	436.870	
IPF Superdome Cell Board (sx1000)	A6866A	1	16,000	32	512.000	
3 Year Svc & Support Price (Hardware and Software)						$1,622,611
4GB SDRAM (4x1GB DIMMS)	A6863A	1	13,898	128	1,778,944	
PCI-x I/O chassis	A6864A	1	16,805	32	537.760	
Core I/O Card	A6865A	1	1,045	2	2.090	
CPU Itanium 2, 1.5GHz w/6MB iL 3 cache (2 CPUs)	A6924A	1	38,000	64	2.432.000	
PCI 1000BT Lan Adapter	A6847A, Opt. 0D1	1	1,325	2	2,650	
HP PCI-X 2-port 4X Fabric (HCA) Adapter	AB286A	1	2,835	8	22,680	
HP 24-port 4X Fabric Copper Switch	AB399A	1	11,800	1	11.800	
HP 5m 4x Fabric Copper Cable	AB346A	1	275	4	1.100	
I/O chassis enclosure for PCI chassis	A5862A	1	25,725	8	205.800	
Graphite I/O expansion power subsystem	A5861D	1	34,860	4	139,440	
PCI 2GB Fibre Channel Adapter	A6795A	1	2,195	176	386,320	
PCI Ultra160 SCSI Adapter	A6828A	1	1,049	2	2.098	
HP Surestore Disk System 2100	A5675A	1	699	2	1.398	
1-36GB LP 10K LVD SE U320 HDD	A6571A	1	651	8	5.208	
HP Rack System/E, 41U	A4902D	1	1,910	2	3,820	
Modular Power Dist Unit for std racks	A5137AZ	1	145	2	290	
200-240 volts North America	A5137AZ	1	94	2	188	
TA5300 Enclosure for DAT tape	C7508AZ	1	1,045	2	2.090	
DDS 4 tape	C7497B	1	1,049	2	2.098	
DVD Rom drive	C7499A	1	515	2	1,030	
SCSI Terminator LVD/SE HDTS68 Multimedia	C2364A	1	100	2	200	
HP Tape Array PSU/Fan Kit	C7496A	1	319	2	638	
SCSI Cable 10m VHDTS68/DHTS68 M/M Multimdedia	C2363B	1	335	4	1.340	
SCSI Cable 0.5m HDTS68 M/M Multimedia	C2978B	1	99	2	198	
SX1000 Superdome SMS, rack	A9802A	1	5,140	2	10,280	
1U Rackmt Display/Keyboard/Mouse	AB243AZ	1	3,046	1	3,046	
				Subtotal	**6,915,056**	**1,622,611**
Server Software						
HPUX 11i, V2 Foundation Operating Environment	B9429AC	1	2,370	128	303.360	
HPUX Fndn OE Media	B9106AA, Opt OD1	1	199	2	398	
				Subtotal	**303,758**	**0**
Storage						
HP Disk Array XP128 SSP Solution	A7875A	1	0	22	0	
XP128 Disk Control Frame	A7876A	1	55,880	22	1.229.360	
HP Proactive 24 Support Upgrade - 3 yrs		1				936,859
XP1024/128 8 Port 1-2GB/sec FC/CA CHIP Pair	A7912B	1	52,490	44	2,309,560	
XP1024/128 2GB Cache Memory Module	A7918A	1	23,170	154	3.568.180	
XP1024/128 512MB Shared Memory Module	A7921A	1	7,170	110	788.700	
XP1024/128 Array Control Processor (ACP)	A7922A	1	58,845	44	2.589.180	
XP128 Disk Path Expansion Kit	A7894A	1	19,190	22	422,180	
XP128 73GB 10k rpm, FC Array Group(4 disks)	A7929A	1	10,417	352	3.666.784	
XP128 73GB 10K rpm, FC Disk Drives	A7929S	1	2,590	44	113.960	
HP ProCurve Switch 2124	A7929S	1	279	1	279	
LUN Conf/Sec Mgr XP 1TB LTU (up to 1TB)	T1614AA	1	13,666	22	300,652	
LUN Conf/Sec Mgr XP 1TB LTU (2-6TB)	T1614AB	1	6,309	88	555,192	
Command View XP LTU	B9357AJ	1	24,690	2	49.380	
16m Fibre Channel Cables	C7525A	1	260	176	45.760	
				Subtotal	**15,639,167**	**936,859**
				Total	**22,857,981**	**2,559,470**
Large Configuration Discount and Support Prepayment*					(11,771,146)	(1,423,259)
				Grand Total	**11,086,835**	**1,136,211**

All the components in the price list are currently available. Maintenance support price is for 24 hours, 7 days with 4 hour response time.

From: MaryBeth Pierantoni [mailto:mary.beth.pierantoni@oracle.com]
Sent: Monday October 4, 2004
To: lucille_boushey@hp.com
Subject: Oracle Pricing for Oracle/HP SuperdomeTPCH Benchmark

Oracle Database 10g Enterprise Edition, Named User Plus for 3 years:	**1,280,000**
Real Application Clusters, Named User Plus for 3 years:	**640,000**
Partitioning, Named User Plus for 3 years:	**320,000**
Oracle Database Server Support Package for 3 years:	**12,000**
Oracle Mandatory E-Business Discount:	**$563,000>**
TOTAL:	**$1,689,000**

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