HP Integrity rx8640
using
Windows 2003 Datacenter Edition (64-bit) SP1
and
Oracle Database 10g Release 2 Enterprise Edition with Partitioning

TPC Benchmark™ H
Full Disclosure Report

First Edition

May 14th, 2007

First Edition – May 14th, 2007

Overview

This report documents the methodology and results of the TPC Benchmark™ H test conducted on the HP Integrity rx8640 , in conformance with the requirements of the TPC Benchmark™ H Standard Specification, Revision 2.6.0. The operating system used for the benchmark was Windows 2003 Datacenter Edition (64-bit) SP1; the DBMS was Oracle 10g Release 2.

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 Lorna Livingtree, Performance Metrics, 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. 3000 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.

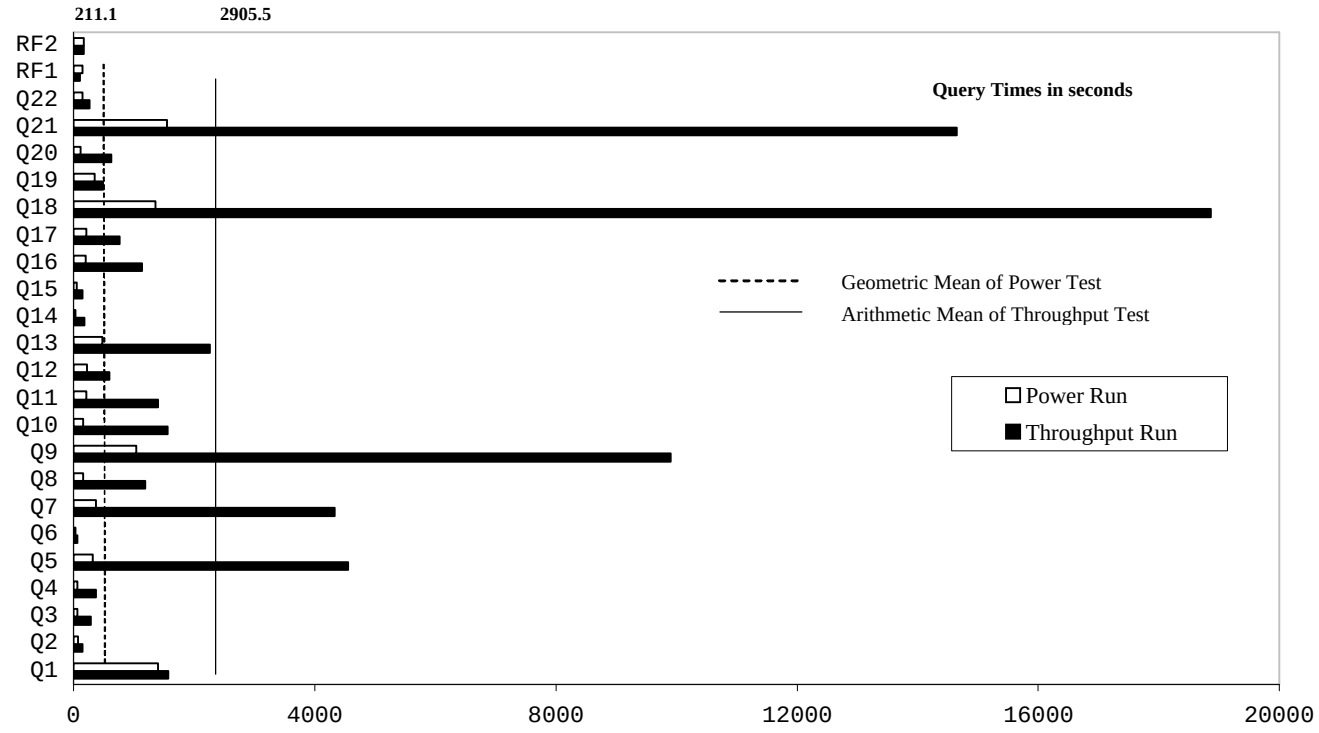
HP Integrity rx8640

TPC-H Rev 2.6.0

Report Date: May 14th, 2007

Total System Cost	Composite Query per Hour Metric	Price/Performance
USD 1,433,521	**37,813.7** QphH@3000GB	**USD 38** QphH@3000GB

Database Size	Database Manager	Operating System	Other Software	Availability Date
3000 GB*	**Oracle Database 10g Release 2 Enterprise Edition with Partitioning**	**Windows 2003 Datacenter Edition (64-bit) SP1**	**MS Visual C++**	**14-May-07**



Database Load Time = 14:15:47	Load Includes Backup: N	Total Data Storage/Database Size = 7.70
RAID (Base Tables Only): N	RAID (Base Tables and Auxiliary Data Structures): N	RAID (All): Y

System Configuration

Processors:	16 Dual-core Intel Itanium2 Processors 9050 1.6GHz, 24MB
Memory:	256 GB
Disk Drives:	2 Internal disks (36GB each) disks plus 64 HP StorageWorks MSA30 (with a total of 640 36GB 15K RPM disks)
Total Disk Storage	23112GB (In this calculation one GB is defined as 1024*1024*1024 bytes)
Lan Controllers	1 PCI 1000BT Lan Adapter

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

HP Integrity rx8640

TPC-H Rev 2.6.0

Report Date: May 14th, 2007

Description	Part Number	Source	Reference Price	Qty	Extended Price	3 yr. Maint. Price
Server Hardware						
rx8640 16-way SMP Base System, incl (4) cell boards, (1) core IO, (6) power supplies, 16x1.6GHz/24MB cache		1				
DC Itanium II processors	AB446A #003		$478,108	1	$478,108	
HP Integrity rx8640 Core I/O Card	AB314A	1	$5,600	1	$5,600	
3 Year Support Price (Hardware and Software)	HA1103A	1	$169,081	1		$169,081
HP Integrity rx8640 I/O Backplane	AD160A	1	$7,700	2	$15,400	
HP 8GB (2x4GB) HD SyncDRAM Midrange Memory	AB455A	1	$16,000	32	$512,000	
HP Integrity rx8640 Sys. Expansion Unit	AB301A	1	$35,785	1	$35,785	
36GB, 15K hard disk for rx76/86	AD146A	1	$1,200	2	$2,400	
DVD+RW Drive for rx76/86	AB351B	1	$850	1	$850	
HP Smart Array Controller 6402	A9890A		$1,669	32	$53,408	
HP Rack Kit for rx86xx Server	J1528B	1	$582	1	$582	
HP Rack Kit for SEU Server	J1530C	1	$709	1	$709	
HP Storageworks MSA 30 SB Storage	302969-B21	1	$2,829	64	$181,056	
HP Storageworks MSA 30 SB Storage 10% Spares	302969-B21	1	$2,829	6	$16,974	
36GB, 15krpm Ultra320 Pluggable Hard Drive	286776-B22	1	$269	640	$172,160	
36GB, 15krpm Ultra320 Pluggable Hard Drive 10% Spar	286776-B22	1	$269	64	$17,216	
146GB, 10krpm U320 Pluggable Hard drive	286716-B22	1	$409	2	$818	
146GB, 10krpm U320 Pluggable Hard drive 10% Spares	286716-B22	1	$409	2	$818	
HP Universal Rack 10642 G2	AF002A		$1,489	6	$8,934	
HP 16A High Voltage Modular PDU	252663-B24	1	$299	9	$2,691	
ML110 G3 SATA / 17" Monitor	www.hp.com	1	$1,076	1	$1,076	
				Subtotal	**1,506,585**	**169,081**
Server Software						
Microsoft Windows Server 2003, Datacenter edition (64-bit)	T2372A opt016	1	$32,000	1	$32,000	
Microsoft Visual C++ Standard	254-00170	1	$109	1	$109	
Oracle Database 10g Release 2 Enterprise Edition		2	$10,000	16*	$160,000	
Named User Plus for 3 years		2				
Partitioning, Named User Plus for 3 years		2	$2,500	16*	$40,000	
Oracle Database Server Support Package for 3 years:		2	$2,000	3		$6,000
				Subtotal	**232,109**	**6,000**
				Total	**1,738,694**	**175,081**
Oracle Mandatory E-Business Discount on (Licenses and Support)					-$30,900	
Large Configuration Discount and Support Prepayment*					(449,354)	0
				Grand Total	**1,258,440**	**175,081**

Source: 1= HP Sales, 19111 Pruneridge Ave., Cupertino, CA 95014 (408) 447 2320
2=Oracle pricing contact: MaryBeth Pierantoni, mary.beth.pierantoni@oracle.com, 916-315-5081

3-yr Cost of Ownership: USD 1,433,521

QphH@3000GB: 37,813.7

$/QphH@3000GB: USD 37.92

* 16 = 0.50 * 32. Explanation: For the purposes of counting the number of processors which require licensing, an Intel multicore chip with "n" cores shall be determined by multiplying "n" cores by a factor of 0.50.

** These components are not immediately orderable. See the FDR for more information.

A 26.31% discount was based on the overall value of the specific components from HP (Price Key) in this single quotation. Discounts for similarly sized configurations will be similar to those quoted here, but may vary based on the components in the quotation.

Audited By: Lorna Livingtree for Performance Metrics Inc

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 rx8640

TPC-H Rev 2.6.0

Report Date: May 14th, 2007

Measurement Results

Database Scaling (SF/size)	3000
Total Data Storage/Database Size	7.70
Start of Database Load Time	5/5 18:09:49
End of Database Load Time	5/6 08:25:36
Database Load Time	14:15:47
Query Streams for Throughput Test (S)	8
TPC-H Power	51,160.6
TPC-H Throughput	27,948.8
TPC-H Composite Query-per-Hour Metric (QphH@3000G	37,813.7
Total System Price Over 3 Years	USD 1,433,521
TPC-H Price/Performance Metric ($/QphH@3000GB)	USD 38

Measurement Intervals

Measurement Interval in Throughput Test (Ts)	68,010

Duration of Stream Execution:

	Seed	Query Start Date/Time Query End Date/Time		RF1 Start Date/Time RF1 End Date/Time		RF2 Start Date/Time RF2 End Date/Time		Duration
Stream 00	506082536	7-May-07 7-May-07	16:43:49 19:13:14	7-May-07 7-May-07	16:39:55 16:43:49	7-May-07 7-May-07	19:13:14 19:16:04	2:32:15
Stream 01	506082537	7-May-07 8-May-07	19:16:12 13:32:55	8-May-07 8-May-07	13:32:55 13:34:47	8-May-07 8-May-07	13:34:47 13:37:35	18:21:23
Stream 02	506082538	7-May-07 8-May-07	19:16:12 12:52:20	8-May-07 8-May-07	13:37:35 13:39:23	8-May-07 8-May-07	13:39:23 13:42:16	18:26:04
Stream 03	506082539	7-May-07 8-May-07	19:16:12 13:03:50	8-May-07 8-May-07	13:42:16 13:44:00	8-May-07 8-May-07	13:44:00 13:46:53	18:30:41
Stream 04	506082540	7-May-07 8-May-07	19:16:12 12:41:08	8-May-07 8-May-07	13:46:53 13:48:34	8-May-07 8-May-07	13:48:34 13:51:27	18:35:15
Stream 05	506082541	7-May-07 8-May-07	19:16:12 12:54:19	8-May-07 8-May-07	13:51:27 13:53:07	8-May-07 8-May-07	13:53:07 13:55:59	18:39:47
Stream 06	506082542	7-May-07 8-May-07	19:16:12 13:02:03	8-May-07 8-May-07	13:55:59 13:57:42	8-May-07 8-May-07	13:57:42 14:00:36	18:44:24
Stream 07	506082543	7-May-07 8-May-07	19:16:12 13:02:06	8-May-07 8-May-07	14:00:36 14:02:15	8-May-07 8-May-07	14:02:15 14:05:09	18:48:57
Stream 08	506082544	7-May-07 8-May-07	19:16:12 13:03:48	8-May-07 8-May-07	14:05:09 14:06:51	8-May-07 8-May-07	14:06:51 14:09:42	18:53:30

HP Integrity rx8640

TPC-H Rev 2.6.0

Report Date May 14th, 2007

TPC-H Timing Intervals (in seconds)

Duration of stream execution:

Query	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Stream 0	1402.2	77.4	64.3	65.4	332.5	29.0	358.1	157.2
Stream 1	1405.3	128.3	217.6	74.4	4294.9	89.9	7538.4	491.8
Stream 2	1595.9	103.6	177.7	332.8	2378.8	121.4	4454.7	444.3
Stream 3	1534.1	221.3	78.5	520.5	4353.8	64.0	2565.4	3349.6
Stream 4	1706.5	157.4	204.5	180.4	6902.1	52.1	3089.6	536.1
Stream 5	1701.0	200.6	165.0	269.8	4449.2	65.9	3732.0	1261.7
Stream 6	1527.2	190.0	611.0	256.2	4766.4	63.1	3629.6	965.1
Stream 7	1526.8	166.1	321.4	230.5	4782.7	64.1	3088.3	556.4
Stream 8	2026.7	160.6	195.6	497.3	4402.7	47.3	3636.2	934.4
Min Qi	1405.3	103.6	78.5	74.4	2378.8	47.3	2565.4	444.3
Max Qi	2026.7	221.3	611.0	520.5	6902.1	121.4	7538.4	3349.6
Avg Qi	1627.9	166.0	246.4	295.2	4541.3	71.0	3966.8	1067.4
Query	**Q9**	**Q10**	**Q11**	**Q12**	**Q13**	**Q14**	**Q15a**	**Q16**
Stream 0	1180.8	157.8	203.1	310.9	476.7	37.6	51.3	201.9
Stream 1	8670.9	1209.0	2708.7	861.4	2732.5	348.5	121.2	601.4
Stream 2	8834.6	1316.7	3124.1	939.7	1740.5	334.4	93.9	960.9
Stream 3	7140.3	1169.6	2450.8	633.9	1640.4	107.6	93.6	586.3
Stream 4	7920.0	926.0	1163.9	519.9	1541.2	200.9	141.9	525.9
Stream 5	8847.4	1012.4	683.3	1071.0	3131.0	301.5	181.7	1053.3
Stream 6	7157.9	4119.8	1061.2	418.5	2780.7	136.7	105.3	2025.9
Stream 7	13846.2	1140.7	837.8	293.2	2157.2	104.9	112.0	580.3
Stream 8	7525.3	941.7	712.2	399.2	1941.3	168.4	181.0	1194.0
Min Qi	7140.3	926.0	683.3	293.2	1541.2	104.9	93.6	525.9
Max Qi	13846.2	4119.8	3124.1	1071.0	3131.0	348.5	181.7	2025.9
Avg Qi	8742.8	1479.5	1592.8	642.1	2208.1	212.8	128.8	941.0
Query	**Q17**	**Q18**	**Q19**	**Q20**	**Q21**	**Q22**	**RF1**	**RF2**
Stream 0	214.4	1469.8	356.2	107.3	1560.0	150.8	234.0	170.0
Stream 1	559.8	15900.2	473.4	726.9	16498.4	150.3	112.8	168.0
Stream 2	2714.1	16577.5	408.8	223.7	16238.3	252.4	108.0	173.1
Stream 3	582.2	16344.9	400.4	208.2	19762.0	250.2	103.5	172.6
Stream 4	398.6	24187.7	390.6	261.3	11430.2	259.0	101.2	173.1
Stream 5	473.7	17140.1	454.8	450.4	16499.6	342.3	100.5	171.9
Stream 6	654.9	16605.2	397.2	679.3	15498.9	301.5	102.2	174.3
Stream 7	368.9	21273.0	484.5	619.4	11129.0	270.9	99.5	173.4
Stream 8	532.7	25442.1	685.8	447.3	11775.6	208.2	101.9	171.5
Min Qi	368.9	15900.2	390.6	208.2	11129.0	150.3	99.5	168.0
Max Qi	2714.1	25442.1	685.8	726.9	19762.0	342.3	112.8	174.3
Avg Qi	785.6	19183.8	461.9	452.1	14854.0	254.4	103.7	172.2

Mr. Mario Broodbakker
Senior Performance Engineer
Hewlett-Packard Company
14475 24th Street NE
Bellevue, WA 98007

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

Platform: **HP Integrity rx8640**
Database Manager: **Oracle 10G Release 2 Enterprise Edition**
Operating System: **Microsoft Windows Server 2003 Datacenter Edition (64bit)**

CPU's	Memory	Total Disks	Qpph@ 3000GB	QthH@3000GB	QphH@3000GB
16 Intel dual core ia64 @ 1.6 Ghz	**256 GB**	**642 @ 36 GB**	51,160.6	27,948.8	37,813.7

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

- **The database tables were defined with the proper columns, layout and sizes.**
- **The tested database was correctly scaled and populated for 3000 GB using DBgen. The version of DBgen was 2.6.0.**
- **The sample data produce by DBgen was successfully compared to the reference data for this scale factor.**
- **The qualification database layout was identical to the tested database except for the number and size of the files.**
- **The query text was verified to use only compliant variants and minor modifications.**
- **The executable query text was generated by Qgen and submitted through Oracle's standard interface. The version of Qgen was 2.6.0. The sample parameters**

produced by Qgen were successfully compared to the reference data for this scale factor.

- **The validation of the query text against the qualification database produced compliant results.**
- **The refresh functions were properly implemented and executed the correct number of inserts and deletes.**
- **The load timing was properly measured and reported.**
- **The execution times were correctly measured and reported.**
- **The performance metrics were correctly computed and reported.**
- **The repeatability of the measurement was verified.**
- **The ACID properties were tested and verified.**
- **The system pricing was checked for major components and maintenance.**
- **The executive summary pages of the FDR were verified for accuracy.**

Auditor's Notes: None

Sincerely,

Lorna Livingtree
Auditor

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 Windows and Oracle Database 10g Release 2 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.

Priced and Measured Configuration:

- 16 1.6GHz Dual-core Intel Itanium2 Processors 9050 CPUs each with 24MB
- 256 GB Memory
- 2 36GB 15K Ultra 320 SCSI Internal disks
- 32 Dual channel Smart Array Controller 6402 Cards
- 1 HP 1000 BaseSX PCI Lan Adapters
- 64 HP StorageWorks MSA30 (with a total of 640 36GB Disks)
- 1 DVD/RW
- 1 SCSI Card

HP Integrity rx8640 Server

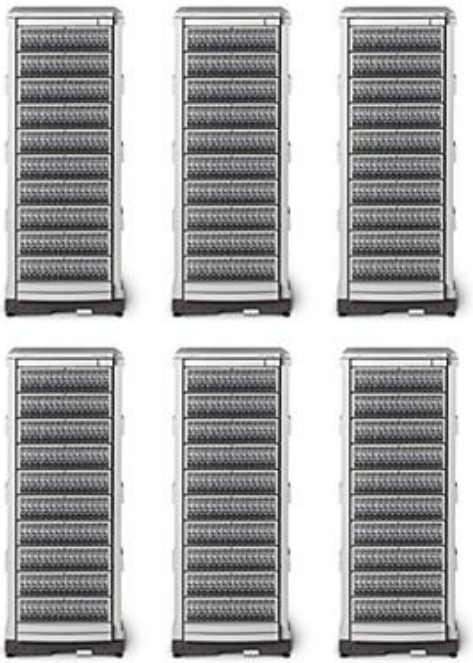
summary

WITH:

- **16 – 1.6 GHz\18MB Dual-Core Itanium2 Processors 9050**
- **256GB Memory**
- **32 PCI Smart Array Controllers (dual-port)**
- **64 HP StorageWorks MSA 30 Enclosures**
- **1 HP 1000 BaseSX PCI Lan Adapter**
- **2 36 GB Ultra 320 SCSI Low Profile Disk (15k)**
- **1 HP Integrity rx8640 Sys. Expansion Unit**
- **1 DVD+RW Drive**

64 HP StorageWorks MSA30

with 640 36GB 15k 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 except for the partsupp table which is an created as an index organised table. 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 2.6.0 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 2.6.0 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.

Variant 15a was used for Query 15.

- 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 Q21 against the qualification database, was started using a randomly selected DELTA.
2. An ACID Transaction, T2, was started for a randomly selected O_KEY, L_KEY and DELTA.
3. T2 completed and appropriate rows in the ORDERS, LINEITEM and HISTORY tables had been changed.
4. Transaction T1 completed executing Q21.

4.5 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 5 protected disk groups. During the durability test, one disk was removed from RAID groups containing the data and the log. The test continued uninterrupted, because of the RAID protection.

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 the server 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	4,500,000,000
LINEITEM	18,000,048,306
CUSTOMER	450,000,000
PART	600,000,000
SUPPLIER	30,000,000
PARTSUPP	2,400,000,000
NATION	25
REGION	5

5.2 Distribution of Tables and Logs Across Media

Distribution of tables and logs across media:

Each MSA30 array (with 10 disks) was configured into 2 Raid5 luns.

LUN1 for TPCH/Oracle ASM use.
LUN2 for flat files and redo log use.

64 luns, one from each MSA30 array, were allocated for Oracle ASM use and a single disk group was built across all 64 luns. All tables, indexes, temp space and other Oracle files were configured in this disk group.

The Windows 2003 OS and the Oracle home directory were configured on two internal disks.

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.

RAID5 was used for log, data, temp, index, and all other files.

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 2.6.0 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 14:15:47.

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
Internal	2	36	72
64 HP StorageWorks MSA30	640	36	23,040.0
TOTAL			**23,112.0**
Scale Factor			**3,000**
Storage Ratio			**7.70**

5.8 Database Load Mechanism Details and Illustration

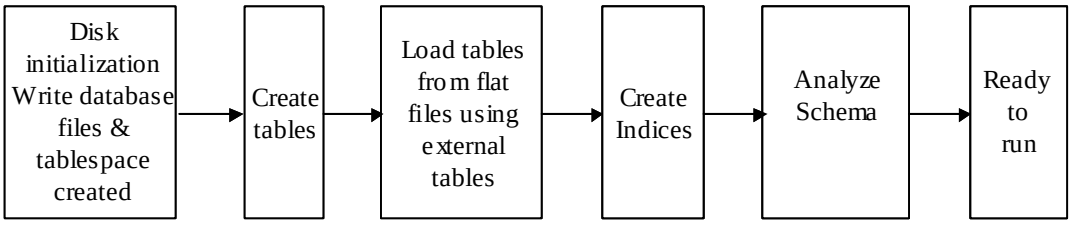
The details of the database load must be described, including a block diagram illustrating the overall process.

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.

8 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.

The Numerical Quantities Summary earlier in this document contain the start and finish times for the refresh functions of each stream.

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@3000GB	QthH@3000GB	QphH@3000GB
Reported Run	51,160.6	27,948.8	37,813.7
Reproducibility Run	53,465.3	28,237.8	38,855.4
% Difference	4.5%	1.0%	2.8%

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 was 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.

All hardware and software is available now

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 disclosure 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 Lorna Livingtree for Performance Metrics. Further information regarding the audit process may be obtained from:

Lorna Livingtree
Performance Metrics Inc.
PO Box 984
Klamath, CA 95548
(707) 482 0523

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

Appendix A
Parameter Settings

A.1 inittpch.ora

```
optimizer_mode = CHOOSE
optimizer_features_enable=10.2.0.2.1
optimizer_index_cost_adj = 1400
optimizer_dynamic_sampling = 3

parallel_adaptive_multi_user = true
parallel_execution_message_size = 16384
parallel_max_servers = 1024
parallel_min_servers = 1024

pga_aggregate_target     = 51539607552


statistics_level = basic
streams_pool_size = 2G
shared_pool_size = 12G
db_cache_size= 50G
log_buffer = 67108864

instance_type = RDBMS
DB_CREATE_FILE_DEST ='+dg1'
aq_tm_processes = 0
audit_trail = false
compatible = 10.2.0.2
control_files = (+dg1/control_1)
db_block_checksum = false
db_block_size = 16384
db_file_multiblock_read_count = 64
db_files = 500
db_name = tpch
dml_locks = 5000
global_names = false

log_checkpoints_to_alert = true
max_dump_file_size = unlimited
nls_date_format = YYYY-MM-DD
open_cursors = 600
processes = 5000
query_rewrite_enabled = true
recovery_parallelism = 64
replication_dependency_tracking = false

timed_statistics = false

undo_management = auto
undo_retention = 400000
undo_tablespace=ts_undo

job_queue_processes = 0
disk_asynch_io = true

background_dump_dest='C:\oracle\product\10.2.0\admin\tpch\bdump'
core_dump_dest='C:\oracle\product\10.2.0\admin\tpch\cdump'
user_dump_dest='C:\oracle\product\10.2.0\admin\tpch\udump'
remote_login_passwordfile='exclusive'
```

A.2 initASM.ora

```
INSTANCE_TYPE = ASM
db_unique_name = ASM
service_names = ASM
instance_name = ASM
large_pool_size = 200M
ASM_DISKSTRING = 'e:\mnt\db\*'
processes = 200
background_dump_dest='C:\oracle\product\10.2.0\admin\+ASM\bdump'
core_dump_dest='C:\oracle\product\10.2.0\admin\+ASM\cdump'
user_dump_dest='C:\oracle\product\10.2.0\admin\+ASM\udump'
remote_login_passwordfile='SHARED'
asm_diskgroups = dg1
```

A.3 Registry

```
HKLM\SYSTEM\CurrentControlSet\Control\
Session Manager]
EnableCMC=0
EnableCPE=0

HKLM\SYSTEM\CurrentControlSet\
Control\Session Manager\Power
IdleFrom0Delay=0

HKLM\SYSTEM\CurrentControlSet\
Control\Session Manager\I/O System
CountOperations=0

HKLM\SYSTEM\CurrentControlSet\
Control\Session Manager\kernel
BackingStoreSecurityMode=0

HKLM\CurrentControlSet\Control\
Session Manager\Memory Management
LargeSystemCache=0

HKLM\SOFTWARE\ORACLE\KEY_OraDb10g_home2
ORA_LPENABLE=1
ORA_LPSIZE=64
```

A.4 env

```
########## MACHINE PARAMETERS ####################
########## PATHS #################################
export SCHEMA_DIR=$KIT_DIR/schema
export PERL=/usr/bin/perl
export BUMPX_DIR=$KIT_DIR/bumpx
export BUMPX_OUT=$KIT_DIR/bumpx
export UTILS=$KIT_DIR/utils
export TEST_DB=/df
export QUAL_DB=$TEST_DB
export DBGEN=$KIT_DIR/dbgen
export ACID_DIR=$KIT_DIR/acid
export QEXEC=$KIT_DIR/utils
export QUERIES=$KIT_DIR/queries
export ANSWERS=$KIT_DIR/answers
export ACID_OUT=$ACID_DIR/acid_out
export ANS2VAL=$ACID_OUT/ans2val
export DSS_CONFIG=$DBGEN
```

```
export DSS_QUERY=$KIT_DIR/queries
export DSS_PATH=$ADE_VIEW_ROOT
export MAINT=$KIT_DIR/maintenance
export CC=gcc
export REGR_TEST=$KIT_DIR/internal/regression_test
export SCALE_FACTOR=3000
export
REF_DATA_SET_DIR=/home/oracle/kit/audit/refdataset/TPCH250_sf3
000
UPDATE_DOP_1=32
UPDATE_DOP_2=64
############ FRAME STUFF
export FRAME=$FRAME_PATH
export ORACORE3INCL=$ORACLE_HOME/rdbms/include
export ORACORE3PUBL=$ORACLE_HOME/rdbms/public
export RDBMSPUBL=$ORACLE_HOME/rdbms/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}:${MAINT}:${ACI
D_DIR}:${FRAME}/bin:${FRAME}/bin:${REGR_TEST}:${PATH}
#
########### ENVIRONMENT VARIABLES ##################
export WORKLOAD=TPCH
export HOSTNAME=orakona
export OPTLEVEL=X02
export GETOPT=-DSTDLIB_HAS_GETOPT
export PLATFORM=

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

########### RULES - do not change these #############
case "$SCALE_FACTOR" in
  1) export NUM_STREAMS=2;;
  10) export NUM_STREAMS=3;;
  100) export NUM_STREAMS=5;;
  300) export NUM_STREAMS=6;;
  1000) export NUM_STREAMS=7;;
  3000) export NUM_STREAMS=8;;
  10000) export NUM_STREAMS=9;;
esac
export DATABASE_USER=tpch/tpch
```

A.5 profile

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
                . ~/.bashrc
fi

# User specific environment and startup programs
export ORACLE_HOME=/u01/app/oracle
export TWO_TASK=tpch
export DBUSER='oraroot/oraroot as sysdba'
export KIT_DIR=/home/oracle/kit
export FRAME_PATH=/home/oracle/frame

PATH=.:$PATH:$HOME/bin:$ORACLE_HOME/bin:$FRAME_PATH
/bin:$KIT_DIR/utils

export LD_LIBRARY_PATH=$ORACLE_HOME/lib


export PATH
unset USERNAME
umask 022
```

Appendix B Build Programs and Scripts

B.1 Create_tpch_db.sql

```
spool cr_tpch_db.log
rem startup nomount;
create database
  controlfile reuse
  logfile 'y:\redolog1' size 40G reuse,
        'y:\redolog2' size 40G reuse
  datafile '+dg1' size 1024m reuse
  sysaux datafile '+dg1' size 1024m reuse
  undo tablespace ts_undo datafile '+dg1' size 60G reuse
  default temporary tablespace ts_temp tempfile '+dg1' size 50G reuse
     extent management local uniform size 5m
  maxdatafiles 400;
alter tablespace ts_undo add datafile '+dg1' size 60G;
alter tablespace ts_undo add datafile '+dg1' size 60G;
alter tablespace ts_undo add datafile '+dg1' size 60G;
alter tablespace ts_undo add datafile '+dg1' size 60G;

set echo off

@?/rdbms/admin/catalog.sql;
@?/rdbms/admin/catproc.sql;

grant sysdba,connect to oraroot identified by oraroot;

grant DBA to tpch identified by tpch;

set spool off
quit;
```

B.2 Tscre_10gR2.sh

```
#!/bin/ksh
echo "START: tablespace creation"
date;

sqlplus  /NOLOG <<!
connect oraroot/oraroot as sysdba

create tablespace ts_default
datafile '+dg1' size 50g reuse
extent management local autoallocate;

create tablespace ts_data nologging
datafile '+dg1' size 60000m reuse
extent management dictionary
default storage (initial 400M next 40m maxextents unlimited pctincrease
0);

create tablespace ts_data2 nologging
datafile '+dg1' size 25000m reuse
extent management dictionary
default storage (initial 100M next 10m maxextents unlimited pctincrease
0);

quit;
!

i=1
while [ $i -lt 64 ]
do
      i=`expr $i + 1`
      addts.sh  ts_data +dg1 60000m &
done

wait;

i=1
while [ $i -lt 64 ]
do
      i=`expr $i + 1`
      addts.sh  ts_data2 +dg1 25000m &
done

wait;

i=1
while [ $i -lt 64 ]
do
      i=`expr $i + 1`
      addtts.sh  ts_temp +dg1 50000m &
done

wait;

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

Addts.sh

```
sqlplus /NOLOG<<!
connect oraroot/oraroot as sysdba;
alter tablespace $1 add datafile '$2' size $3 reuse;
!
```

Addtts.sh

```
sqlplus /NOLOG<<!
connect oraroot/oraroot as sysdba;
alter tablespace $1 add tempfile '$2' size $3 reuse;
!
```

B.3 Dapop_10gR2.sh

```
#!/bin/bash
echo "START load"
date
sqlplus /NOLOG <<EOF
connect oraroot/oraroot as sysdba;

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

connect tpch/tpch;

set timing on
set echo on

create or replace directory ff1 as 'y:';
create or replace directory ff2 as 'y:';
create or replace directory ff3 as 'y:';
create or replace directory ff4 as 'y:';
create or replace directory ff5 as 'y:';
create or replace directory ff6 as 'y:';
```

```
create or replace directory ff7 as 'y:';
create or replace directory ff8 as 'y:';
create or replace directory ff9 as 'y:';
create or replace directory ff10 as 'y:';
create or replace directory ff11 as 'y:';
create or replace directory ff12 as 'y:';
create or replace directory ff13 as 'y:';
create or replace directory ff14 as 'y:';
create or replace directory ff15 as 'y:';
create or replace directory ff16 as 'y:';
create or replace directory ff17 as 'y:';
create or replace directory ff18 as 'y:';
create or replace directory ff19 as 'y:';
create or replace directory ff20 as 'y:';
create or replace directory ff21 as 'y:';
create or replace directory ff22 as 'y:';
create or replace directory ff23 as 'y:';
create or replace directory ff24 as 'y:';
create or replace directory ff25 as 'y:';
create or replace directory ff26 as 'y:';
create or replace directory ff27 as 'y:';
create or replace directory ff28 as 'y:';
create or replace directory ff29 as 'y:';
create or replace directory ff30 as 'y:';
create or replace directory ff31 as 'y:';
create or replace directory ff32 as 'y:';
create or replace directory ff33 as 'x:';
create or replace directory ff34 as 'x:';
create or replace directory ff35 as 'x:';
create or replace directory ff36 as 'x:';
create or replace directory ff37 as 'x:';
create or replace directory ff38 as 'x:';
create or replace directory ff39 as 'x:';
create or replace directory ff40 as 'x:';
create or replace directory ff41 as 'x:';
create or replace directory ff42 as 'x:';
create or replace directory ff43 as 'x:';
create or replace directory ff44 as 'x:';
create or replace directory ff45 as 'x:';
create or replace directory ff46 as 'x:';
create or replace directory ff47 as 'x:';
create or replace directory ff48 as 'x:';
create or replace directory ff49 as 'x:';
create or replace directory ff50 as 'x:';
create or replace directory ff51 as 'x:';
create or replace directory ff52 as 'x:';
create or replace directory ff53 as 'x:';
create or replace directory ff54 as 'x:';
create or replace directory ff55 as 'x:';
create or replace directory ff56 as 'x:';
create or replace directory ff57 as 'x:';
create or replace directory ff58 as 'x:';
create or replace directory ff59 as 'x:';
create or replace directory ff60 as 'x:';
create or replace directory ff61 as 'x:';
create or replace directory ff62 as 'x:';
create or replace directory ff63 as 'x:';
create or replace directory ff64 as 'x:';

drop table l_et;
create table l_et(
   l_orderkey          number ,
   l_partkey           number ,
   l_suppkey           number ,
   l_linenumber        number ,
   l_quantity          number ,
   l_extendedprice     number ,
   l_discount          number ,
   l_tax               number ,
   l_returnflag        char(1) ,
   l_linestatus        char(1) ,
   l_shipdate          date ,
   l_commitdate        date ,
   l_receiptdate       date ,
   l_shipinstruct      char(25) ,
   l_shipmode          char(10) ,
   l_comment           varchar(44)
)
organization external (
type ORACLE_LOADER
default directory ff1
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
                )
                location (
ff1:'lineitem.tbl.1',
ff2:'lineitem.tbl.2',
ff3:'lineitem.tbl.3',
ff4:'lineitem.tbl.4',
ff5:'lineitem.tbl.5',
ff6:'lineitem.tbl.6',
ff7:'lineitem.tbl.7',
ff8:'lineitem.tbl.8',
ff9:'lineitem.tbl.9',
ff10:'lineitem.tbl.10',
ff11:'lineitem.tbl.11',
ff12:'lineitem.tbl.12',
ff13:'lineitem.tbl.13',
ff14:'lineitem.tbl.14',
ff15:'lineitem.tbl.15',
ff16:'lineitem.tbl.16',
ff17:'lineitem.tbl.17',
ff18:'lineitem.tbl.18',
ff19:'lineitem.tbl.19',
ff20:'lineitem.tbl.20',
ff21:'lineitem.tbl.21',
ff22:'lineitem.tbl.22',
ff23:'lineitem.tbl.23',
ff24:'lineitem.tbl.24',
ff25:'lineitem.tbl.25',
ff26:'lineitem.tbl.26',
ff27:'lineitem.tbl.27',
ff28:'lineitem.tbl.28',
ff29:'lineitem.tbl.29',
ff30:'lineitem.tbl.30',
ff31:'lineitem.tbl.31',
ff32:'lineitem.tbl.32',
ff33:'lineitem.tbl.33',
ff34:'lineitem.tbl.34',
ff35:'lineitem.tbl.35',
ff36:'lineitem.tbl.36',
ff37:'lineitem.tbl.37',
ff38:'lineitem.tbl.38',
ff39:'lineitem.tbl.39',
ff40:'lineitem.tbl.40',
ff41:'lineitem.tbl.41',
ff42:'lineitem.tbl.42',
ff43:'lineitem.tbl.43',
ff44:'lineitem.tbl.44',
ff45:'lineitem.tbl.45',
ff46:'lineitem.tbl.46',
ff47:'lineitem.tbl.47',
ff48:'lineitem.tbl.48',
ff49:'lineitem.tbl.49',
ff50:'lineitem.tbl.50',
ff51:'lineitem.tbl.51',
ff52:'lineitem.tbl.52',
```

```
ff53:'lineitem.tbl.53',
ff54:'lineitem.tbl.54',
ff55:'lineitem.tbl.55',
ff56:'lineitem.tbl.56',
ff57:'lineitem.tbl.57',
ff58:'lineitem.tbl.58',
ff59:'lineitem.tbl.59',
ff60:'lineitem.tbl.60',
ff61:'lineitem.tbl.61',
ff62:'lineitem.tbl.62',
ff63:'lineitem.tbl.63',
ff64:'lineitem.tbl.64'
))
reject limit unlimited;

drop table o_et;
create table o_et(
   o_orderkey          number ,
   o_custkey           number ,
   o_orderstatus       char(1) ,
   o_totalprice        number ,
   o_orderdate         date ,
   o_orderpriority     char(15) ,
   o_clerk             char(15) ,
   o_shippriority      number ,
   o_comment           varchar(79)
)
organization external (
type ORACLE_LOADER
default directory ff1
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
            )
            location (
ff1:'orders.tbl.1',
ff2:'orders.tbl.2',
ff3:'orders.tbl.3',
ff4:'orders.tbl.4',
ff5:'orders.tbl.5',
ff6:'orders.tbl.6',
ff7:'orders.tbl.7',
ff8:'orders.tbl.8',
ff9:'orders.tbl.9',
ff10:'orders.tbl.10',
ff11:'orders.tbl.11',
ff12:'orders.tbl.12',
ff13:'orders.tbl.13',
ff14:'orders.tbl.14',
ff15:'orders.tbl.15',
ff16:'orders.tbl.16',
ff17:'orders.tbl.17',
ff18:'orders.tbl.18',
ff19:'orders.tbl.19',
ff20:'orders.tbl.20',
ff21:'orders.tbl.21',
ff22:'orders.tbl.22',
ff23:'orders.tbl.23',
ff24:'orders.tbl.24',
ff25:'orders.tbl.25',
ff26:'orders.tbl.26',
ff27:'orders.tbl.27',
ff28:'orders.tbl.28',
ff29:'orders.tbl.29',
ff30:'orders.tbl.30',
ff31:'orders.tbl.31',
ff32:'orders.tbl.32',
ff33:'orders.tbl.33',
ff34:'orders.tbl.34',
ff35:'orders.tbl.35',
ff36:'orders.tbl.36',
ff37:'orders.tbl.37',
ff38:'orders.tbl.38',
ff39:'orders.tbl.39',
ff40:'orders.tbl.40',
ff41:'orders.tbl.41',
ff42:'orders.tbl.42',
ff43:'orders.tbl.43',
ff44:'orders.tbl.44',
ff45:'orders.tbl.45',
ff46:'orders.tbl.46',
ff47:'orders.tbl.47',
ff48:'orders.tbl.48',
ff49:'orders.tbl.49',
ff50:'orders.tbl.50',
ff51:'orders.tbl.51',
ff52:'orders.tbl.52',
ff53:'orders.tbl.53',
ff54:'orders.tbl.54',
ff55:'orders.tbl.55',
ff56:'orders.tbl.56',
ff57:'orders.tbl.57',
ff58:'orders.tbl.58',
ff59:'orders.tbl.59',
ff60:'orders.tbl.60',
ff61:'orders.tbl.61',
ff62:'orders.tbl.62',
ff63:'orders.tbl.63',
ff64:'orders.tbl.64'
))
reject limit unlimited;

drop table ps_et;
create table ps_et(
   ps_partkey          number ,
   ps_suppkey          number ,
   ps_availqty         number ,
   ps_supplycost       number ,
   ps_comment          varchar(199)
)
organization external (
type ORACLE_LOADER
default directory ff1
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
            )
            location (
ff1:'partsupp.tbl.1',
ff2:'partsupp.tbl.2',
ff3:'partsupp.tbl.3',
ff4:'partsupp.tbl.4',
ff5:'partsupp.tbl.5',
ff6:'partsupp.tbl.6',
ff7:'partsupp.tbl.7',
ff8:'partsupp.tbl.8',
ff9:'partsupp.tbl.9',
ff10:'partsupp.tbl.10',
ff11:'partsupp.tbl.11',
ff12:'partsupp.tbl.12',
ff13:'partsupp.tbl.13',
ff14:'partsupp.tbl.14',
ff15:'partsupp.tbl.15',
ff16:'partsupp.tbl.16',
```

```
ff17:'partsupp.tbl.17',
ff18:'partsupp.tbl.18',
ff19:'partsupp.tbl.19',
ff20:'partsupp.tbl.20',
ff21:'partsupp.tbl.21',
ff22:'partsupp.tbl.22',
ff23:'partsupp.tbl.23',
ff24:'partsupp.tbl.24',
ff25:'partsupp.tbl.25',
ff26:'partsupp.tbl.26',
ff27:'partsupp.tbl.27',
ff28:'partsupp.tbl.28',
ff29:'partsupp.tbl.29',
ff30:'partsupp.tbl.30',
ff31:'partsupp.tbl.31',
ff32:'partsupp.tbl.32',
ff33:'partsupp.tbl.33',
ff34:'partsupp.tbl.34',
ff35:'partsupp.tbl.35',
ff36:'partsupp.tbl.36',
ff37:'partsupp.tbl.37',
ff38:'partsupp.tbl.38',
ff39:'partsupp.tbl.39',
ff40:'partsupp.tbl.40',
ff41:'partsupp.tbl.41',
ff42:'partsupp.tbl.42',
ff43:'partsupp.tbl.43',
ff44:'partsupp.tbl.44',
ff45:'partsupp.tbl.45',
ff46:'partsupp.tbl.46',
ff47:'partsupp.tbl.47',
ff48:'partsupp.tbl.48',
ff49:'partsupp.tbl.49',
ff50:'partsupp.tbl.50',
ff51:'partsupp.tbl.51',
ff52:'partsupp.tbl.52',
ff53:'partsupp.tbl.53',
ff54:'partsupp.tbl.54',
ff55:'partsupp.tbl.55',
ff56:'partsupp.tbl.56',
ff57:'partsupp.tbl.57',
ff58:'partsupp.tbl.58',
ff59:'partsupp.tbl.59',
ff60:'partsupp.tbl.60',
ff61:'partsupp.tbl.61',
ff62:'partsupp.tbl.62',
ff63:'partsupp.tbl.63',
ff64:'partsupp.tbl.64'
))
reject limit unlimited;

drop table p_et;
create table p_et(
   p_partkey          number ,
   p_name             varchar(55) ,
   p_mfgr             char(25) ,
   p_brand            char(10) ,
   p_type             varchar(25) ,
   p_size             number ,
   p_container        char(10) ,
   p_retailprice      number ,
   p_comment          varchar(23)
)
organization external (
type ORACLE_LOADER
default directory ff1
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
                )
                location (
ff1:'part.tbl.1',
ff2:'part.tbl.2',
ff3:'part.tbl.3',
ff4:'part.tbl.4',
ff5:'part.tbl.5',
ff6:'part.tbl.6',
ff7:'part.tbl.7',
ff8:'part.tbl.8',
ff9:'part.tbl.9',
ff10:'part.tbl.10',
ff11:'part.tbl.11',
ff12:'part.tbl.12',
ff13:'part.tbl.13',
ff14:'part.tbl.14',
ff15:'part.tbl.15',
ff16:'part.tbl.16',
ff17:'part.tbl.17',
ff18:'part.tbl.18',
ff19:'part.tbl.19',
ff20:'part.tbl.20',
ff21:'part.tbl.21',
ff22:'part.tbl.22',
ff23:'part.tbl.23',
ff24:'part.tbl.24',
ff25:'part.tbl.25',
ff26:'part.tbl.26',
ff27:'part.tbl.27',
ff28:'part.tbl.28',
ff29:'part.tbl.29',
ff30:'part.tbl.30',
ff31:'part.tbl.31',
ff32:'part.tbl.32',
ff33:'part.tbl.33',
ff34:'part.tbl.34',
ff35:'part.tbl.35',
ff36:'part.tbl.36',
ff37:'part.tbl.37',
ff38:'part.tbl.38',
ff39:'part.tbl.39',
ff40:'part.tbl.40',
ff41:'part.tbl.41',
ff42:'part.tbl.42',
ff43:'part.tbl.43',
ff44:'part.tbl.44',
ff45:'part.tbl.45',
ff46:'part.tbl.46',
ff47:'part.tbl.47',
ff48:'part.tbl.48',
ff49:'part.tbl.49',
ff50:'part.tbl.50',
ff51:'part.tbl.51',
ff52:'part.tbl.52',
ff53:'part.tbl.53',
ff54:'part.tbl.54',
ff55:'part.tbl.55',
ff56:'part.tbl.56',
ff57:'part.tbl.57',
ff58:'part.tbl.58',
ff59:'part.tbl.59',
ff60:'part.tbl.60',
ff61:'part.tbl.61',
ff62:'part.tbl.62',
ff63:'part.tbl.63',
ff64:'part.tbl.64'
))
reject limit unlimited;
```

```
drop table c_et;
create table c_et(
    c_custkey           number ,
    c_name              varchar(25) ,
    c_address           varchar(40) ,
    c_nationkey          number ,
    c_phone             char(15) ,
    c_acctbal           number ,
    c_mktsegment         char(10) ,
    c_comment           varchar(117)
)
organization external (
type ORACLE_LOADER
default directory ff1
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
                )
                location (
ff1:'customer.tbl.1',
ff2:'customer.tbl.2',
ff3:'customer.tbl.3',
ff4:'customer.tbl.4',
ff5:'customer.tbl.5',
ff6:'customer.tbl.6',
ff7:'customer.tbl.7',
ff8:'customer.tbl.8',
ff9:'customer.tbl.9',
ff10:'customer.tbl.10',
ff11:'customer.tbl.11',
ff12:'customer.tbl.12',
ff13:'customer.tbl.13',
ff14:'customer.tbl.14',
ff15:'customer.tbl.15',
ff16:'customer.tbl.16',
ff17:'customer.tbl.17',
ff18:'customer.tbl.18',
ff19:'customer.tbl.19',
ff20:'customer.tbl.20',
ff21:'customer.tbl.21',
ff22:'customer.tbl.22',
ff23:'customer.tbl.23',
ff24:'customer.tbl.24',
ff25:'customer.tbl.25',
ff26:'customer.tbl.26',
ff27:'customer.tbl.27',
ff28:'customer.tbl.28',
ff29:'customer.tbl.29',
ff30:'customer.tbl.30',
ff31:'customer.tbl.31',
ff32:'customer.tbl.32',
ff33:'customer.tbl.33',
ff34:'customer.tbl.34',
ff35:'customer.tbl.35',
ff36:'customer.tbl.36',
ff37:'customer.tbl.37',
ff38:'customer.tbl.38',
ff39:'customer.tbl.39',
ff40:'customer.tbl.40',
ff41:'customer.tbl.41',
ff42:'customer.tbl.42',
ff43:'customer.tbl.43',
ff44:'customer.tbl.44',
ff45:'customer.tbl.45',
ff46:'customer.tbl.46',
ff47:'customer.tbl.47',
ff48:'customer.tbl.48',
ff49:'customer.tbl.49',
ff50:'customer.tbl.50',
ff51:'customer.tbl.51',
ff52:'customer.tbl.52',
ff53:'customer.tbl.53',
ff54:'customer.tbl.54',
ff55:'customer.tbl.55',
ff56:'customer.tbl.56',
ff57:'customer.tbl.57',
ff58:'customer.tbl.58',
ff59:'customer.tbl.59',
ff60:'customer.tbl.60',
ff61:'customer.tbl.61',
ff62:'customer.tbl.62',
ff63:'customer.tbl.63',
ff64:'customer.tbl.64'
))
reject limit unlimited;

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

```
ff31:'supplier.tbl.31',
ff32:'supplier.tbl.32',
ff33:'supplier.tbl.33',
ff34:'supplier.tbl.34',
ff35:'supplier.tbl.35',
ff36:'supplier.tbl.36',
ff37:'supplier.tbl.37',
ff38:'supplier.tbl.38',
ff39:'supplier.tbl.39',
ff40:'supplier.tbl.40',
ff41:'supplier.tbl.41',
ff42:'supplier.tbl.42',
ff43:'supplier.tbl.43',
ff44:'supplier.tbl.44',
ff45:'supplier.tbl.45',
ff46:'supplier.tbl.46',
ff47:'supplier.tbl.47',
ff48:'supplier.tbl.48',
ff49:'supplier.tbl.49',
ff50:'supplier.tbl.50',
ff51:'supplier.tbl.51',
ff52:'supplier.tbl.52',
ff53:'supplier.tbl.53',
ff54:'supplier.tbl.54',
ff55:'supplier.tbl.55',
ff56:'supplier.tbl.56',
ff57:'supplier.tbl.57',
ff58:'supplier.tbl.58',
ff59:'supplier.tbl.59',
ff60:'supplier.tbl.60',
ff61:'supplier.tbl.61',
ff62:'supplier.tbl.62',
ff63:'supplier.tbl.63',
ff64:'supplier.tbl.64'
))
reject limit unlimited;

drop table n_et;
create table n_et(
   n_nationkey          number ,
   n_name               char(25) ,
   n_regionkey          number ,
   n_comment            varchar(152)
)
organization external (
type ORACLE_LOADER
default directory ff1
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
            )
            location (
ff1:'nation.tbl'
))
reject limit unlimited;

drop table r_et;
create table r_et(
   r_regionkey          number ,
   r_name               char(25) ,
   r_comment            varchar(152)
)
organization external (
type ORACLE_LOADER
default directory ff1
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
            )
            location (
ff1:'region.tbl'
))
reject limit unlimited;

drop table r_et;
create table r_et(
   r_regionkey          number ,
   r_name               char(25) ,
   r_comment            varchar(152)
)
organization external (
type ORACLE_LOADER
default directory ff1
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
            )
            location (
ff1:'region.tbl'
))
reject limit unlimited;
!date
drop table l_et2;
create table l_et2(
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 ff33
access parameters
(
records delimited by newline
nobadfile
nologfile
fields terminated by '|'
missing field values are null
)
location (ff33:'lineitem.tbl.642'))
reject limit unlimited;
!date
drop table o_et2;

create table o_et2(
o_orderkey number ,
o_custkey number ,
```

```
o_orderstatus char(1) ,
o_totalprice number ,
o_orderdate date ,
o_orderpriority char(15) ,
o_clerk char(15) ,
o_shippriority number ,
o_comment varchar(79)
)
organization external (
type ORACLE_LOADER
default directory ff1
access parameters
(
records delimited by newline
nobadfile
nologfile
fields terminated by '|'
missing field values are null
)
location (ff33:'orders.tbl.642'))
reject limit unlimited;


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

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

@?/rdbms/admin/utlxplan.sql;
!date
set timing on
set echo on
create table lineitem(
   l_shipdate          ,
   l_orderkey          NOT NULL,
   l_discount          NOT NULL,
   l_extendedprice     NOT NULL,
   l_suppkey           NOT NULL,
   l_quantity          NOT NULL,
   l_returnflag        ,
   l_partkey           NOT NULL,
   l_linestatus        ,
   l_tax               NOT NULL,
   l_commitdate        ,
   l_receiptdate       ,
   l_shipmode          ,
   l_linenumber        NOT NULL,
   l_shipinstruct      ,
   l_comment
)
pctfree 1
pctused 99
initrans 10
tablespace ts_data
storage (initial 320m next 50m freelist groups 4 freelists 99)
compress
parallel
nologging
partition by range (l_shipdate)
subpartition by hash(l_partkey)
subpartitions 64
(
partition item1 values less than (to_date('1992-01-01','YYYY-MM-
DD')) ,
partition item2 values less than (to_date('1992-02-01','YYYY-MM-
DD')) ,
partition item3 values less than (to_date('1992-03-01','YYYY-MM-
DD')) ,
partition item4 values less than (to_date('1992-04-01','YYYY-MM-
DD')) ,
partition item5 values less than (to_date('1992-05-01','YYYY-MM-
DD')) ,
partition item6 values less than (to_date('1992-06-01','YYYY-MM-
DD')) ,
partition item7 values less than (to_date('1992-07-01','YYYY-MM-
DD')) ,
partition item8 values less than (to_date('1992-08-01','YYYY-MM-
DD')) ,
partition item9 values less than (to_date('1992-09-01','YYYY-MM-
DD')) ,
partition item10 values less than (to_date('1992-10-01','YYYY-MM-
DD')) ,
partition item11 values less than (to_date('1992-11-01','YYYY-MM-
DD')) ,
partition item12 values less than (to_date('1992-12-01','YYYY-MM-
DD')) ,
partition item13 values less than (to_date('1993-01-01','YYYY-MM-
DD')) ,
partition item14 values less than (to_date('1993-02-01','YYYY-MM-
DD')) ,
partition item15 values less than (to_date('1993-03-01','YYYY-MM-
DD')) ,
partition item16 values less than (to_date('1993-04-01','YYYY-MM-
DD')) ,
partition item17 values less than (to_date('1993-05-01','YYYY-MM-
DD')) ,
partition item18 values less than (to_date('1993-06-01','YYYY-MM-
DD')) ,
partition item19 values less than (to_date('1993-07-01','YYYY-MM-
DD')) ,
partition item20 values less than (to_date('1993-08-01','YYYY-MM-
DD')) ,
partition item21 values less than (to_date('1993-09-01','YYYY-MM-
DD')) ,
partition item22 values less than (to_date('1993-10-01','YYYY-MM-
DD')) ,
partition item23 values less than (to_date('1993-11-01','YYYY-MM-
DD')) ,
partition item24 values less than (to_date('1993-12-01','YYYY-MM-
DD')) ,
partition item25 values less than (to_date('1994-01-01','YYYY-MM-
DD')) ,
partition item26 values less than (to_date('1994-02-01','YYYY-MM-
DD')) ,
partition item27 values less than (to_date('1994-03-01','YYYY-MM-
DD')) ,
partition item28 values less than (to_date('1994-04-01','YYYY-MM-
DD')) ,
partition item29 values less than (to_date('1994-05-01','YYYY-MM-
DD')) ,
partition item30 values less than (to_date('1994-06-01','YYYY-MM-
DD')) ,
partition item31 values less than (to_date('1994-07-01','YYYY-MM-
DD')) ,
partition item32 values less than (to_date('1994-08-01','YYYY-MM-
DD')) ,
partition item33 values less than (to_date('1994-09-01','YYYY-MM-
DD')) ,
partition item34 values less than (to_date('1994-10-01','YYYY-MM-
DD')) ,
partition item35 values less than (to_date('1994-11-01','YYYY-MM-
DD')) ,
partition item36 values less than (to_date('1994-12-01','YYYY-MM-
DD')) ,
partition item37 values less than (to_date('1995-01-01','YYYY-MM-
DD')) ,
```

```
partition item38 values less than (to_date('1995-02-01','YYYY-MM-
DD')) ,
partition item39 values less than (to_date('1995-03-01','YYYY-MM-
DD')) ,
partition item40 values less than (to_date('1995-04-01','YYYY-MM-
DD')) ,
partition item41 values less than (to_date('1995-05-01','YYYY-MM-
DD')) ,
partition item42 values less than (to_date('1995-06-01','YYYY-MM-
DD')) ,
partition item43 values less than (to_date('1995-07-01','YYYY-MM-
DD')) ,
partition item44 values less than (to_date('1995-08-01','YYYY-MM-
DD')) ,
partition item45 values less than (to_date('1995-09-01','YYYY-MM-
DD')) ,
partition item46 values less than (to_date('1995-10-01','YYYY-MM-
DD')) ,
partition item47 values less than (to_date('1995-11-01','YYYY-MM-
DD')) ,
partition item48 values less than (to_date('1995-12-01','YYYY-MM-
DD')) ,
partition item49 values less than (to_date('1996-01-01','YYYY-MM-
DD')) ,
partition item50 values less than (to_date('1996-02-01','YYYY-MM-
DD')) ,
partition item51 values less than (to_date('1996-03-01','YYYY-MM-
DD')) ,
partition item52 values less than (to_date('1996-04-01','YYYY-MM-
DD')) ,
partition item53 values less than (to_date('1996-05-01','YYYY-MM-
DD')) ,
partition item54 values less than (to_date('1996-06-01','YYYY-MM-
DD')) ,
partition item55 values less than (to_date('1996-07-01','YYYY-MM-
DD')) ,
partition item56 values less than (to_date('1996-08-01','YYYY-MM-
DD')) ,
partition item57 values less than (to_date('1996-09-01','YYYY-MM-
DD')) ,
partition item58 values less than (to_date('1996-10-01','YYYY-MM-
DD')) ,
partition item59 values less than (to_date('1996-11-01','YYYY-MM-
DD')) ,
partition item60 values less than (to_date('1996-12-01','YYYY-MM-
DD')) ,
partition item61 values less than (to_date('1997-01-01','YYYY-MM-
DD')) ,
partition item62 values less than (to_date('1997-02-01','YYYY-MM-
DD')) ,
partition item63 values less than (to_date('1997-03-01','YYYY-MM-
DD')) ,
partition item64 values less than (to_date('1997-04-01','YYYY-MM-
DD')) ,
partition item65 values less than (to_date('1997-05-01','YYYY-MM-
DD')) ,
partition item66 values less than (to_date('1997-06-01','YYYY-MM-
DD')) ,
partition item67 values less than (to_date('1997-07-01','YYYY-MM-
DD')) ,
partition item68 values less than (to_date('1997-08-01','YYYY-MM-
DD')) ,
partition item69 values less than (to_date('1997-09-01','YYYY-MM-
DD')) ,
partition item70 values less than (to_date('1997-10-01','YYYY-MM-
DD')) ,
partition item71 values less than (to_date('1997-11-01','YYYY-MM-
DD')) ,
partition item72 values less than (to_date('1997-12-01','YYYY-MM-
DD')) ,
partition item73 values less than (to_date('1998-01-01','YYYY-MM-
DD')) ,
partition item74 values less than (to_date('1998-02-01','YYYY-MM-
DD')) ,
partition item75 values less than (to_date('1998-03-01','YYYY-MM-
DD')) ,
partition item76 values less than (to_date('1998-04-01','YYYY-MM-
DD')) ,
partition item77 values less than (to_date('1998-05-01','YYYY-MM-
DD')) ,
partition item78 values less than (to_date('1998-06-01','YYYY-MM-
DD')) ,
partition item79 values less than (to_date('1998-07-01','YYYY-MM-
DD')) ,
partition item80 values less than (to_date('1998-08-01','YYYY-MM-
DD')) ,
partition item81 values less than (to_date('1998-09-01','YYYY-MM-
DD')) ,
partition item82 values less than (to_date('1998-10-01','YYYY-MM-
DD')) ,
partition item83 values less than (to_date('1998-11-01','YYYY-MM-
DD')) ,
partition item84 values less than (MAXVALUE))
as select
   l_shipdate          ,
   l_orderkey          ,
   l_discount          ,
   l_extendedprice     ,
   l_suppkey           ,
   l_quantity          ,
   l_returnflag        ,
   l_partkey           ,
   l_linestatus        ,
   l_tax               ,
   l_commitdate        ,
   l_receiptdate       ,
   l_shipmode          ,
   l_linenumber        ,
   l_shipinstruct      ,
   l_comment
from l_et order by l_orderkey;
!date

!date
create table orders(
   o_orderdate         ,
   o_orderkey          NOT NULL,
   o_custkey           NOT NULL,
   o_orderpriority     ,
   o_shippriority      ,
   o_clerk             ,
   o_orderstatus       ,
   o_totalprice        ,
   o_comment
)
pctfree 1
pctused 99
initrans 10
tablespace ts_data
storage (initial 128m next 20m freelist groups 4 freelists 99)
compress
parallel
nologging
partition by range (o_orderdate)
subpartition by hash(o_custkey)
subpartitions 64
(
partition ord1 values less than (to_date('1992-01-01','YYYY-MM-DD')),
partition ord2 values less than (to_date('1992-02-01','YYYY-MM-DD')),
partition ord3 values less than (to_date('1992-03-01','YYYY-MM-DD')),
partition ord4 values less than (to_date('1992-04-01','YYYY-MM-DD')),
```

```
partition ord5 values less than (to_date('1992-05-01','YYYY-MM-DD')),
partition ord6 values less than (to_date('1992-06-01','YYYY-MM-DD')),
partition ord7 values less than (to_date('1992-07-01','YYYY-MM-DD')),
partition ord8 values less than (to_date('1992-08-01','YYYY-MM-DD')),
partition ord9 values less than (to_date('1992-09-01','YYYY-MM-DD')),
partition ord10 values less than (to_date('1992-10-01','YYYY-MM-
DD')),
partition ord11 values less than (to_date('1992-11-01','YYYY-MM-
DD')),
partition ord12 values less than (to_date('1992-12-01','YYYY-MM-
DD')),
partition ord13 values less than (to_date('1993-01-01','YYYY-MM-
DD')),
partition ord14 values less than (to_date('1993-02-01','YYYY-MM-
DD')),
partition ord15 values less than (to_date('1993-03-01','YYYY-MM-
DD')),
partition ord16 values less than (to_date('1993-04-01','YYYY-MM-
DD')),
partition ord17 values less than (to_date('1993-05-01','YYYY-MM-
DD')),
partition ord18 values less than (to_date('1993-06-01','YYYY-MM-
DD')),
partition ord19 values less than (to_date('1993-07-01','YYYY-MM-
DD')),
partition ord20 values less than (to_date('1993-08-01','YYYY-MM-
DD')),
partition ord21 values less than (to_date('1993-09-01','YYYY-MM-
DD')),
partition ord22 values less than (to_date('1993-10-01','YYYY-MM-
DD')),
partition ord23 values less than (to_date('1993-11-01','YYYY-MM-
DD')),
partition ord24 values less than (to_date('1993-12-01','YYYY-MM-
DD')),
partition ord25 values less than (to_date('1994-01-01','YYYY-MM-
DD')),
partition ord26 values less than (to_date('1994-02-01','YYYY-MM-
DD')),
partition ord27 values less than (to_date('1994-03-01','YYYY-MM-
DD')),
partition ord28 values less than (to_date('1994-04-01','YYYY-MM-
DD')),
partition ord29 values less than (to_date('1994-05-01','YYYY-MM-
DD')),
partition ord30 values less than (to_date('1994-06-01','YYYY-MM-
DD')),
partition ord31 values less than (to_date('1994-07-01','YYYY-MM-
DD')),
partition ord32 values less than (to_date('1994-08-01','YYYY-MM-
DD')),
partition ord33 values less than (to_date('1994-09-01','YYYY-MM-
DD')),
partition ord34 values less than (to_date('1994-10-01','YYYY-MM-
DD')),
partition ord35 values less than (to_date('1994-11-01','YYYY-MM-
DD')),
partition ord36 values less than (to_date('1994-12-01','YYYY-MM-
DD')),
partition ord37 values less than (to_date('1995-01-01','YYYY-MM-
DD')),
partition ord38 values less than (to_date('1995-02-01','YYYY-MM-
DD')),
partition ord39 values less than (to_date('1995-03-01','YYYY-MM-
DD')),
partition ord40 values less than (to_date('1995-04-01','YYYY-MM-
DD')),
partition ord41 values less than (to_date('1995-05-01','YYYY-MM-
DD')),
partition ord42 values less than (to_date('1995-06-01','YYYY-MM-
DD')),
partition ord43 values less than (to_date('1995-07-01','YYYY-MM-
DD')),
partition ord44 values less than (to_date('1995-08-01','YYYY-MM-
DD')),
partition ord45 values less than (to_date('1995-09-01','YYYY-MM-
DD')),
partition ord46 values less than (to_date('1995-10-01','YYYY-MM-
DD')),
partition ord47 values less than (to_date('1995-11-01','YYYY-MM-
DD')),
partition ord48 values less than (to_date('1995-12-01','YYYY-MM-
DD')),
partition ord49 values less than (to_date('1996-01-01','YYYY-MM-
DD')),
partition ord50 values less than (to_date('1996-02-01','YYYY-MM-
DD')),
partition ord51 values less than (to_date('1996-03-01','YYYY-MM-
DD')),
partition ord52 values less than (to_date('1996-04-01','YYYY-MM-
DD')),
partition ord53 values less than (to_date('1996-05-01','YYYY-MM-
DD')),
partition ord54 values less than (to_date('1996-06-01','YYYY-MM-
DD')),
partition ord55 values less than (to_date('1996-07-01','YYYY-MM-
DD')),
partition ord56 values less than (to_date('1996-08-01','YYYY-MM-
DD')),
partition ord57 values less than (to_date('1996-09-01','YYYY-MM-
DD')),
partition ord58 values less than (to_date('1996-10-01','YYYY-MM-
DD')),
partition ord59 values less than (to_date('1996-11-01','YYYY-MM-
DD')),
partition ord60 values less than (to_date('1996-12-01','YYYY-MM-
DD')),
partition ord61 values less than (to_date('1997-01-01','YYYY-MM-
DD')),
partition ord62 values less than (to_date('1997-02-01','YYYY-MM-
DD')),
partition ord63 values less than (to_date('1997-03-01','YYYY-MM-
DD')),
partition ord64 values less than (to_date('1997-04-01','YYYY-MM-
DD')),
partition ord65 values less than (to_date('1997-05-01','YYYY-MM-
DD')),
partition ord66 values less than (to_date('1997-06-01','YYYY-MM-
DD')),
partition ord67 values less than (to_date('1997-07-01','YYYY-MM-
DD')),
partition ord68 values less than (to_date('1997-08-01','YYYY-MM-
DD')),
partition ord69 values less than (to_date('1997-09-01','YYYY-MM-
DD')),
partition ord70 values less than (to_date('1997-10-01','YYYY-MM-
DD')),
partition ord71 values less than (to_date('1997-11-01','YYYY-MM-
DD')),
partition ord72 values less than (to_date('1997-12-01','YYYY-MM-
DD')),
partition ord73 values less than (to_date('1998-01-01','YYYY-MM-
DD')),
partition ord74 values less than (to_date('1998-02-01','YYYY-MM-
DD')),
partition ord75 values less than (to_date('1998-03-01','YYYY-MM-
DD')),
partition ord76 values less than (to_date('1998-04-01','YYYY-MM-
DD')),
partition ord77 values less than (to_date('1998-05-01','YYYY-MM-
DD')),
```

```
partition ord78 values less than (to_date('1998-06-01','YYYY-MM-
DD')),
partition ord79 values less than (to_date('1998-07-01','YYYY-MM-
DD')),
partition ord80 values less than (to_date('1998-08-01','YYYY-MM-
DD')),
partition ord81 values less than (to_date('1998-09-01','YYYY-MM-
DD')),
partition ord82 values less than (to_date('1998-10-01','YYYY-MM-
DD')),
partition ord83 values less than (to_date('1998-11-01','YYYY-MM-
DD')),
partition ord84 values less than (MAXVALUE))
as select
   o_orderdate         ,
   o_orderkey          ,
   o_custkey           ,
   o_orderpriority     ,
   o_shippriority      ,
   o_clerk             ,
   o_orderstatus       ,
   o_totalprice        ,
   o_comment
from o_et order by o_orderkey;
!date

create table partsupp(
ps_partkey NOT NULL ,
ps_suppkey NOT NULL ,
ps_supplycost NOT NULL ,
ps_availqty,
ps_comment
, constraint pk_partkey_suppkey_1 primary
key(ps_partkey,ps_suppkey)
)
organization index
pctthreshold 50
storage (initial 720M)
tablespace ts_data2
compress
parallel
nologging
partition by hash (ps_partkey)
partitions 64
as select
ps_partkey,
ps_suppkey,
ps_supplycost,
ps_availqty,
ps_comment
from ps_et;

create table customer(
   c_custkey           NOT NULL,
   c_mktsegment        ,
   c_nationkey         ,
   c_name              ,
   c_address           ,
   c_phone             ,
   c_acctbal           ,
   c_comment
)
pctfree 0
pctused 99
storage (freelist groups 4 freelists 99)
compress
parallel
nologging
partition by hash (c_custkey)
partitions 64
tablespace ts_data2
as select
   c_custkey           ,
   c_mktsegment        ,
   c_nationkey         ,
   c_name              ,
   c_address           ,
   c_phone             ,
   c_acctbal           ,
   c_comment
from c_et;
!date

create table part(
   p_partkey           NOT NULL,
   p_type              ,
   p_size              ,
   p_brand             ,
   p_name              ,
   p_container         ,
   p_mfgr              ,
   p_retailprice       ,
   p_comment
)
pctfree 0
pctused 99
storage (freelist groups 4 freelists 99)
compress
parallel
nologging
partition by hash (p_partkey)
partitions 64
tablespace ts_data2
as select
   p_partkey           ,
   p_type              ,
   p_size              ,
   p_brand             ,
   p_name              ,
   p_container         ,
   p_mfgr              ,
   p_retailprice       ,
   p_comment
from p_et;
!date

create table supplier(
   s_suppkey           NOT NULL,
   s_nationkey         ,
   s_comment           ,
   s_name              ,
   s_address           ,
   s_phone             ,
   s_acctbal
)
pctfree 0
pctused 99
storage (freelist groups 4 freelists 99)
compress
parallel
nologging
partition by hash (s_suppkey)
partitions 64
tablespace ts_data2
as select
   s_suppkey           ,
   s_nationkey         ,
   s_comment           ,
   s_name              ,
   s_address           ,
   s_phone             ,
   s_acctbal
```

```
from s_et;
!date

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

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

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

!date

rem drop index i_l_orderkey;
create index i_l_orderkey
on lineitem (l_orderkey) global partition by hash (l_orderkey)
partitions 64
pctfree 2
initrans 10
tablespace ts_data2
storage (freelist groups 4 freelists 99)
parallel
compute statistics
nologging;

!date

rem drop index i_o_orderkey;
create unique index i_o_orderkey
on orders (o_orderkey) global partition by hash (o_orderkey)
partitions 64
pctfree 2
initrans 10
tablespace ts_data2
storage (freelist groups 4 freelists 99 )
parallel
compute statistics
nologging;
!date

rem drop index i_c_custkey;
create unique index i_c_custkey
on customer (c_custkey)  global partition by hash (c_custkey)
partitions 64
pctfree 2
initrans 10
tablespace ts_data2
storage (freelist groups 4 freelists 99)
parallel
compute statistics
nologging;

!date
alter session force parallel dml parallel (degree 64);
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 l_et2) order by l_orderkey;
commit;
insert into orders (
select
o_orderdate ,
o_orderkey ,
o_custkey ,
o_orderpriority ,
o_shippriority ,
o_clerk ,
o_orderstatus ,
o_totalprice ,
o_comment
from o_et2) order by o_orderkey;
commit;

!date
alter index i_l_orderkey allocate extent (size 35M);
alter index i_l_orderkey allocate extent (size 35M);
alter index i_l_orderkey allocate extent (size 35M);
alter index i_l_orderkey allocate extent (size 35M);
alter index i_o_orderkey allocate extent (size 16M);
alter index i_o_orderkey allocate extent (size 16M);
alter index i_o_orderkey allocate extent (size 16M);
alter index i_o_orderkey allocate extent (size 16M);
!date

execute dbms_stats.gather_schema_stats('TPCH' , estimate_percent =>
1, degree => 32 , granularity => 'GLOBAL' ,method_opt => 'for all
columns size 1');

connect oraroot/oraroot as sysdba;

execute dbms_stats.gather_system_stats;
execute dbms_scheduler.disable('GATHER_STATS_JOB');
execute dbms_scheduler.disable('AUTO_SPACE_ADVISOR_JOB');
execute dbms_scheduler.disable('AUTO_TASKS_JOB_CLASS');

alter system switch logfile;
alter system switch logfile;

!date
quit;
EOF
```

B.4 Create_diskgroup.sql (ASM)

```
drop diskgroup dg1 including contents;
CREATE DISKGROUP dg1 EXTERNAL REDUNDANCY DISK
'e:\mnt\db\0' ,
'e:\mnt\db\1' ,
```

'e:\mnt\db\2' ,
'e:\mnt\db\3' ,
'e:\mnt\db\4' ,
'e:\mnt\db\5' ,
'e:\mnt\db\6' ,
'e:\mnt\db\7' ,
'e:\mnt\db\8' ,
'e:\mnt\db\9' ,
'e:\mnt\db\10' ,
'e:\mnt\db\11' ,
'e:\mnt\db\12' ,
'e:\mnt\db\13' ,
'e:\mnt\db\14' ,
'e:\mnt\db\15' ,
'e:\mnt\db\16' ,
'e:\mnt\db\17' ,
'e:\mnt\db\18' ,
'e:\mnt\db\19' ,
'e:\mnt\db\20' ,
'e:\mnt\db\21' ,
'e:\mnt\db\22' ,
'e:\mnt\db\23' ,
'e:\mnt\db\24' ,
'e:\mnt\db\25' ,
'e:\mnt\db\26' ,
'e:\mnt\db\27' ,
'e:\mnt\db\28' ,
'e:\mnt\db\29' ,
'e:\mnt\db\30' ,
'e:\mnt\db\31' ,
'e:\mnt\db\32' ,
'e:\mnt\db\33' ,
'e:\mnt\db\34' ,
'e:\mnt\db\35' ,
'e:\mnt\db\36' ,
'e:\mnt\db\37' ,
'e:\mnt\db\38' ,
'e:\mnt\db\39' ,
'e:\mnt\db\40' ,
'e:\mnt\db\41' ,
'e:\mnt\db\42' ,
'e:\mnt\db\43' ,
'e:\mnt\db\44' ,
'e:\mnt\db\45' ,
'e:\mnt\db\46' ,
'e:\mnt\db\47' ,
'e:\mnt\db\48' ,
'e:\mnt\db\49' ,
'e:\mnt\db\50' ,
'e:\mnt\db\51' ,
'e:\mnt\db\52' ,
'e:\mnt\db\53' ,
'e:\mnt\db\54' ,
'e:\mnt\db\55' ,
'e:\mnt\db\56' ,
'e:\mnt\db\57' ,
'e:\mnt\db\58' ,
'e:\mnt\db\59' ,
'e:\mnt\db\60' ,
'e:\mnt\db\61' ,
'e:\mnt\db\62' ,
'e:\mnt\db\63' ;

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

set serverout on;

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

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

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

exit;
```

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 doatrans;
END;
/
show errors
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(outfile, "ACID Transaction iteration %d ROLLBACK at
%s\n",
          num_iter, ctime(&curr_time));
   }

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

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

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

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

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

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

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

  tr_end = gettime();

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

  /*  Disconnect from ORACLE. */

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

  ACIDexit();

  exit(0);
}


void ACIDinit()
{

  /* run random seed */

  srand48(getpid());

  /* Connect to ORACLE.  Program will call sql_error()
```

```
    if an error occurs in connecting to the default database. */

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

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

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

  /* get username and password */

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

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


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

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

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

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


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcusr,0,OCI_ATTR_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,strlen((char *)sqlstmt),
              OCI_NTV_SYNTAX,OCI_DEFAULT);
  OCIsexec(tpcsvc,curi,errhp,1);*/

  /* Make session serializable */

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

  /* Set optimizer_index_cost_adj = 25 */

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

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


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

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



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

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

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

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

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

#define ISOTXT "alter session set isolation_level = serializable"
#define PDMLTXT "alter session force parallel dml parallel (degree 2)"
#define PDDLTXT "alter session force parallel ddl parallel (degree 2)"
#define OICATXT "alter session set optimizer_index_cost_adj=25"

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

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

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

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

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

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

#endif /* ATRANSPL_H */
```

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 oraroot/oraroot 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 @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
/usr/bin/rsh orakona "type
C:\oracle\product\10.2.0\admin\tpc1\bdump\alert_tpc1.log"  >> $CHK
i=0
while [ $i -lt $STREAM ]
do
KEYS=`head -10 ${KEY}$i | awk '{printf "%d ", $1}'`
#for j in `head -10 ${KEY}$i | awk '{printf "%d ", $1}'`
echo "The keys to check for consistency after the test from file $i are:"
echo "$KEYS"
for j in $KEYS
do
   sqlplus $USER @consist $j >> $CON2
   echo "------------------------------" >> $CON2
done
   i=`expr $i + 1`
done
```

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/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
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
```

C.14 iso.sh

```
#!/bin/ksh
#
# $Header: iso.sh 17-aug-99.15:44:51 mpoess Exp $
#
# iso.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#   NAME
#     iso.sh
#
#   DESCRIPTION
#    This script triggers all 6 isolation tests. In addition,
#    it creates more readable formats of the isolation test output.
#   NOTES
#     <other useful comments, qualifications, etc.>
#
#   MODIFIED   (MM/DD/YY)
#   mpoess      08/17/99 - Creation
#   mpoess      08/17/99 - Creation
#
. $KIT_DIR/env
for iso in iso1 iso2 iso3 iso4 iso5 iso6;do
              echo Running isolation test $iso
              ${iso}.sh
              echo Creating nicely formated output of ACID test $iso
              xiso.pl -o ${ACID_OUT}/${iso}
done
```

C.15 iso1.sh

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

. $KIT_DIR/env

# May need to change the following:
RSH=ssh

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

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

USER=$DATABASE_USER
PROG=atranspl


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

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

usage() {

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

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

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


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

# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

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

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

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

sleep 1

# start ACID transaction, Sleep for 60 second before COMMIT

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

# let's sleep 10 seconds before starting ACID query

sleep 10

# start ACID query with the same OKEY
```

```
echo "Running ACID query 15 seconds AFTER the start of ACID
Transaction" \
>> $TXN2FILE
echo "`date`" >> $TXN2FILE
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >>
$TXN2FILE

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

C.16 iso2.sh

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

. $KIT_DIR/env

RSH=ssh

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

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

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

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

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

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

sleep 1

# start ACID transaction, Sleep for 30 second before
ROLLBACK

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

# let's sleep 15 seconds before starting ACID query

sleep 10

# start ACID query with the same OKEY

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

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

C.17 iso3.sh

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

. $KIT_DIR/env

# May need to change the following:
RSH=ssh

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

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE
#scp $KEYFILE ${HOST}:$KEYFILE  changed by mb 11/08

sleep 1

# start ACID transaction, Sleep for 30 second before COMMIT

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

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

```
sleep 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} -n ${HOST} $PROG 2 2 1 1 i$KEYFILE u$USER s1 b1 >>
$TXN2FILE &
else
$PROG 2 2 1 1 i$KEYFILE u$USER s1 b1 >> $TXN2FILE &
fi

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

C.18 iso4.sh

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

. $KIT_DIR/env

# May need to change the following:
RSH=ssh

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

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

sleep 1

# start ACID transaction, Sleep for 30 second before ROLLBACK

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

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

sleep 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.

echo "Starting TXN2 on node $HOST" >> $TXN2FILE
$PROG 2 2 1 1 i$KEYFILE u$USER s1 b1 >> $TXN2FILE &

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

C.19 iso5.sh

```
#!/bin/ksh
#
# $Header: iso5.sh 04-aug-99.09:21:45 mpoess Exp $
#
# iso5.sh
#
```

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

. $KIT_DIR/env

# May need to change the following:
RSH=ssh

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

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

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 2

# 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

echo "Starting PARTSUPP query on node $HOST" >> $TXN2FILE
sqlplus $USER @$ACID_DIR/isolation/a_query2 ${PSKEY} >>
$TXN2FILE &

wait

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

C.20 iso6.sh

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

```
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/04/99 - Creation
#    mpoess      08/04/99 - Creation
#

. $KIT_DIR/env

# May need to change the following:
RSH=ssh

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

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

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 2

# start Query 1, use 0 as the delta

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

# sleep 2 seconds before starting ACID transaction

sleep 2

# start ACID transaction, COMMIT after one second

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

echo "Starting ACID transaction on node $HOST" >> $TXN2FILE
$PROG 1 1 1 0 i$KEYFILE u$USER s1 >> $TXN2FILE &

# start Query 17

sleep 2

echo "Running 2nd Query 1 at `date`" >> $TXN3FILE
sqlplus $USER @q1 >> $TXN3FILE &
# wait for everyone to finish

wait

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

cat $TXN1FILE $TXN2FILE $TXN3FILE >> $ISOFILE

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

C.21 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
```

C.22 q1.sql

```
Rem
Rem $Header: template.sql 06-feb-96.13:23:14 mpoess   Exp $
```

```
Rem
Rem q1.sql
Rem
Rem  Copyright (c) Oracle Corporation 2001. All Rights Reserved.
Rem
Rem    NAME
Rem      q1.sql - <one-line expansion of the name>
Rem
Rem    DESCRIPTION
Rem      used in isolation test 6
Rem
Rem    NOTES
Rem      <other useful comments, qualifications, etc.>
Rem
Rem    MODIFIED   (MM/DD/YY)
Rem    mpoess      02/13/01 - Created
Rem

set serverout on;

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

select
       l_returnflag,
       l_linestatus,
       sum(l_quantity) as sum_qty,
       sum(l_extendedprice) as sum_base_price,
       sum(l_extendedprice * (1 - l_discount)) as sum_disc_price,
       sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) as
sum_charge,
       avg(l_quantity) as avg_qty,
       avg(l_extendedprice) as avg_price,
       avg(l_discount) as avg_disc,
       count(*) as count_order
from
       lineitem
where
       l_shipdate <= to_date ('1998-12-01','YYYY-MM-DD')  - 0
group by
       l_returnflag,
       l_linestatus
order by
       l_returnflag,
       l_linestatus;

select
'AFTER ACID QUERY Q1' as STAGE,
substr(TO_CHAR(sysdate,'YYYY-MM-DD HH:MI:SS'),1,20) as
CURRENT_TIME
from dual;

exit;
```

C.23 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.24 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.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/env

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

usage() {

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

ITER=600
STEM=${NUM_STREAMS}
let STEM="$STEM + 1" # add one for the update stream
SF=1
PROG=atranspl
IN=${ACID_DIR}/acid_in
DURA_DIR=$ACID_OUT/dura
OUT=$DURA_DIR/drate
DURA=$DURA_DIR/dura
KEY=${DURA_DIR}/key$$_
echo "$$" > ${DURA_DIR}/shellpid
USER=tpch/tpch
TRIG=1
HCNT=duracntb

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

# get all the options

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


echo "Starting ACID run..."

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

# Get history count before the run
```

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

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

wait
# perform the consistency

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

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

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

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

done

wait

echo "ACID run completed"
```

C.26 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.27 sample.sql

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

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

exit;
```

Appendix D Query text and Output

D.1 Query Qualification

-- Q1

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

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


4 rows processed.
Query Processed in 8.88 seconds.
```

-- Q2

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

```
S_ACCTBAL          S_NAME                N_NAME
P_PARTKEY          P_MFGR
S_ADDRESS                         S_PHONE
S_COMMENT
9938.53            Supplier#000005359    UNITED
KINGDOM
```

185358.00 Manufacturer#4 QKuHYh,vZGiwu2FWEJoLDx04 33-429-790-6131
uriously regular requests hag
9937.84 Supplier#000005969 ROMANIA 108438.00 Manufacturer#1 ANDENSOSmk,miq23Xfb5RWt6dvUcvt6Qa 29-520-692-3537
efully express instructions. regular requests against the slyly fin
9936.22 Supplier#000005250 UNITED KINGDOM 249.00 Manufacturer#4 B3rqp0xbSEim4Mpy2RH J 33-320-228-2957
etect about the furiously final accounts. slyly ironic pinto beans sleep inside the furiously
9923.77 Supplier#000002324 GERMANY 29821.00 Manufacturer#4 y3OD9UywSTOk 17-779-299-1839
ackages boost blithely. blithely regular deposits c
9871.22 Supplier#000006373 GERMANY 43868.00 Manufacturer#5 J8fcXWsTqM 17-813-485-8637
etect blithely bold asymptotes. fluffily ironic platelets wake furiously; blit
9870.78 Supplier#000001286 GERMANY 81285.00 Manufacturer#2 YKA,E2fjiVd7eUrzp2Ef8j1QxGo2DFnosaTEH 17-516-924-4574
regular accounts. furiously unusual courts above the fi
9870.78 Supplier#000001286 GERMANY 181285.00 Manufacturer#4 YKA,E2fjiVd7eUrzp2Ef8j1QxGo2DFnosaTEH 17-516-924-4574
regular accounts. furiously unusual courts above the fi
9852.52 Supplier#000008973 RUSSIA 18972.00 Manufacturer#2 t5L67YdBYYH6o,Vz24jpDyQ9 32-188-594-7038
rns wake final foxes. carefully unusual depende
9847.83 Supplier#000008097 RUSSIA 130557.00 Manufacturer#2 xMe97bpE69NzdwLoX 32-375-640-3593
the special excuses. silent sentiments serve carefully final ac
9847.57 Supplier#000006345 FRANCE 86344.00 Manufacturer#1 VSt3rzk3qG698u6ld8HhOByvrTcSTSvQlDQDag 16-886-766-7945
ges. slyly regular requests are. ruthless, express excuses cajole blithely across the unu
9847.57 Supplier#000006345 FRANCE 173827.00 Manufacturer#2 VSt3rzk3qG698u6ld8HhOByvrTcSTSvQlDQDag 16-886-766-7945
ges. slyly regular requests are. ruthless, express excuses cajole blithely across the unu
9836.93 Supplier#000007342 RUSSIA 4841.00 Manufacturer#4 JOlK7C1,7xrEZSSOw 32-399-414-5385
blithely carefully bold theodolites. fur
9817.10 Supplier#000002352 RUSSIA 124815.00 Manufacturer#2 4LfoHUZjgjEbAKw TgdKcgOc4D4uCYw 32-551-831-1437
wake carefully alongside of the carefully final ex
9817.10 Supplier#000002352 RUSSIA 152351.00 Manufacturer#3 4LfoHUZjgjEbAKw TgdKcgOc4D4uCYw 32-551-831-1437
wake carefully alongside of the carefully final ex
9739.86 Supplier#000003384 FRANCE 138357.00 Manufacturer#2 o,Z3v4POifevE k9U1b 6J1ucX,I 16-494-913-5925
s after the furiously bold packages sleep fluffily idly final requests: quickly final
9721.95 Supplier#000008757 UNITED KINGDOM 156241.00 Manufacturer#3 Atg6GnM4dT2 33-821-407-2995
eep furiously sauternes; quickl
9681.33 Supplier#000008406 RUSSIA 78405.00 Manufacturer#1 ,qUuXcftUl 32-139-873-8571
haggle slyly regular excuses. quic
9643.55 Supplier#000005148 ROMANIA 107617.00 Manufacturer#1 kT4ciVFslx9z4s79p Js825 29-252-617-4850
final excuses. final ideas boost quickly furiously speci
9624.82 Supplier#000001816 FRANCE 34306.00 Manufacturer#3 e7vab91vLJPWxxZnewmnDBpDmxYHrb 16-392-237-6726
e packages are around the special ideas. special, pending foxes us
9624.78 Supplier#000009658 ROMANIA 189657.00 Manufacturer#1 oE9uBgEfSS4opIcepXyAYM,x 29-748-876-2014
ronic asymptotes wake bravely final
9612.94 Supplier#000003228 ROMANIA 120715.00 Manufacturer#2 KDdpNKN3cWu7ZSrbdqp7AfSLxx,qWB 29-325-784-8187
warhorses. quickly even deposits sublate daringly ironic instructions. slyly blithe t
9612.94 Supplier#000003228 ROMANIA 198189.00 Manufacturer#4 KDdpNKN3cWu7ZSrbdqp7AfSLxx,qWB 29-325-784-8187
warhorses. quickly even deposits sublate daringly ironic instructions. slyly blithe t
9571.83 Supplier#000004305 ROMANIA

179270.00 Manufacturer#2 qNHZ7WmCzygwMPRDO9Ps 29-973-481-1831
kly carefully express asymptotes. furiou
9558.10 Supplier#000003532 UNITED KINGDOM
88515.00 Manufacturer#4 EOeuiiOn21OVpTlGguufFDFsbN1p0lhpxHp 33-152-301-2164
foxes. quickly even excuses use. slyly special foxes nag bl
9492.79 Supplier#000005975 GERMANY
25974.00 Manufacturer#5 S6mIiCTx82z7lV 17-992-579-4839
arefully pending accounts. blithely regular excuses boost carefully carefully ironic p
9461.05 Supplier#000002536 UNITED KINGDOM
20033.00 Manufacturer#1 8mmGbyzaU 7ZS2wJumTibypncu9pNkDc4FYA 33-556-973-5522
. slyly regular deposits wake slyly. furiously regular warthogs are.
9453.01 Supplier#000000802 ROMANIA
175767.00 Manufacturer#1 ,6HYXb4uaHITmtMBj4Ak57Pd 29-342-882-6463
gular frets. permanently special multipliers believe blithely alongs
9408.65 Supplier#000007772 UNITED KINGDOM
117771.00 Manufacturer#4 AiC5YAH,gdu0i7 33-152-491-1126
nag against the final requests. furiously unusual packages cajole blit
9359.61 Supplier#000004856 ROMANIA
62349.00 Manufacturer#5 HYogcF3Jb yh1 29-334-870-9731
y ironic theodolites. blithely sile
9357.45 Supplier#000006188 UNITED KINGDOM
138648.00 Manufacturer#1 g801,ssP8wpTk4Hm 33-583-607-1633
ously always regular packages. fluffily even accounts beneath the furiously final pack
9352.04 Supplier#000003439 GERMANY
170921.00 Manufacturer#4 qYPDgoiBGhCYxjgC 17-128-996-4650
according to the carefully bold ideas
9312.97 Supplier#000007807 RUSSIA
90279.00 Manufacturer#5 oGYMPCk9XHGB2PBfKRnHA 32-673-872-5854
ecial packages among the pending, even requests use regula
9312.97 Supplier#000007807 RUSSIA
100276.00 Manufacturer#5 oGYMPCk9XHGB2PBfKRnHA 32-673-872-5854
ecial packages among the pending, even requests use regula
9280.27 Supplier#000007194 ROMANIA
47193.00 Manufacturer#3 zhRUQkBSrFYxIAXTfInj vyGRQjeK 29-318-454-2133
o beans haggle after the furiously unusual deposits. carefully silent dolphins cajole carefully
9274.80 Supplier#000008854 RUSSIA
76346.00 Manufacturer#3 1xhLoOUM7I3mZ1mKnerw OSqdbb4QbGa 32-524-148-5221
y. courts do wake slyly. carefully ironic platelets haggle above the slyly regular the
9249.35 Supplier#000003973 FRANCE
26466.00 Manufacturer#1 d18GiDsL6Wm2IsGXM,RZf1jCsgZAOjNYVThTRP4 16-722-866-1658
uests are furiously. regular tithes through the regular, final accounts cajole furiously above the q
9249.35 Supplier#000003973 FRANCE
33972.00 Manufacturer#1 d18GiDsL6Wm2IsGXM,RZf1jCsgZAOjNYVThTRP4 16-722-866-1658
uests are furiously. regular tithes through the regular, final accounts cajole furiously above the q
9208.70 Supplier#000007769 ROMANIA
40256.00 Manufacturer#5 rsimdze 5o9P Ht7xS 29-964-424-9649
lites was quickly above the furiously ironic requests. slyly even foxes against the blithely bold
9201.47 Supplier#000009690 UNITED KINGDOM
67183.00 Manufacturer#5 CB BnUTlmi5zdeEl7R7 33-121-267-9529
e even, even foxes. blithely ironic packages cajole regular packages. slyly final ide
9192.10 Supplier#000000115 UNITED KINGDOM
85098.00 Manufacturer#3 nJ 2t0f7Ve,wL1,6WzGBJLNBUCKlsV 33-597-248-1220
es across the carefully express accounts boost caref
9189.98 Supplier#000001226 GERMANY
21225.00 Manufacturer#4 qsLCqSvLyZfuXIpjz 17-725-903-1381
deposits. blithely bold excuses about the slyly bold forges wake
9128.97 Supplier#000004311 RUSSIA
146768.00 Manufacturer#5 I8IjnXd7NSJRs594RxsRR0 32-155-440-7120
refully. blithely unusual asymptotes haggle
9104.83 Supplier#000008520 GERMANY
150974.00 Manufacturer#4 RqRVDgD0ER J9 b41vR2,3 17-728-804-1793

ly about the blithely ironic depths. slyly final theodolites among the fluffily bold ideas print
9101.00 Supplier#000005791 ROMANIA 128254.00 Manufacturer#5 zub2zCV,jhHPPQqi,P2INAjE1zI n66cOEoXFG 29-549-251-5384
ts. notornis detect blithely above the carefully bold requests. blithely even package
9094.57 Supplier#000004582 RUSSIA 39575.00 Manufacturer#1 WB0XkCSG3r,mnQ n,h9VIxjjr9ARHFvKgMDf 32-587-577-1351
jole. regular accounts sleep blithely frets. final pinto beans play furiously past the
8996.87 Supplier#000004702 FRANCE 102191.00 Manufacturer#5 8XVcQK23akp 16-811-269-8946
ickly final packages along the express plat
8996.14 Supplier#000009814 ROMANIA 139813.00 Manufacturer#2 af0O5pg83lPU4IDVmEylXZVqYZQzSDlYLAmR 29-995-571-8781
dependencies boost quickly across the furiously pending requests! unusual dolphins play sl
8968.42 Supplier#000010000 ROMANIA 119999.00 Manufacturer#5 aTGLEusCiL4F PDBdv665XBJhPyCOB0i 29-578-432-2146
ly regular foxes boost slyly. quickly special waters boost carefully ironi
8936.82 Supplier#000007043 UNITED KINGDOM 109512.00 Manufacturer#1 FVajceZInZdbJE6Z9XsRUxrUEpiwHDrOXi,1Rz 33-784-177-8208
efully regular courts. furiousl
8929.42 Supplier#000008770 FRANCE 173735.00 Manufacturer#4 R7cG26TtXrHAP9 HckhfRi 16-242-746-9248
cajole furiously unusual requests. quickly stealthy requests are.
8920.59 Supplier#000003967 ROMANIA 26460.00 Manufacturer#1 eHoAXe62SY9 29-194-731-3944
aters. express, pending instructions sleep. brave, r
8920.59 Supplier#000003967 ROMANIA 173966.00 Manufacturer#2 eHoAXe62SY9 29-194-731-3944
aters. express, pending instructions sleep. brave, r
8913.96 Supplier#000004603 UNITED KINGDOM 137063.00 Manufacturer#2 OUzlvMUr7n,utLxmPNeYKSf3T24OXskxB5 33-789-255-7342
haggle slyly above the furiously regular pinto beans. even
8877.82 Supplier#000007967 FRANCE 167966.00 Manufacturer#5 A3pi1BARM4nx6R,qrwFoRPU 16-442-147-9345
ously foxes. express, ironic requests im
8862.24 Supplier#000003323 ROMANIA 73322.00 Manufacturer#3 W9 lYcsC9FwBqk3ItL 29-736-951-3710
ly pending ideas sleep about the furiously unu
8841.59 Supplier#000005750 ROMANIA 100729.00 Manufacturer#5 Erx3lAgu0g62iaHF9x50uMH4EgeN9hEG 29-344-502-5481
gainst the pinto beans. fluffily unusual dependencies affix slyly even deposits.
8781.71 Supplier#000003121 ROMANIA 13120.00 Manufacturer#5 wNqTogx238ZYCamFb,50v,bj 4IbNFW9Bvw1xP 29-707-291-5144
s wake quickly ironic ideas
8754.24 Supplier#000009407 UNITED KINGDOM 179406.00 Manufacturer#4 CHRCbkaWcf5B 33-903-970-9604
e ironic requests. carefully even foxes above the furious
8691.06 Supplier#000004429 UNITED KINGDOM 126892.00 Manufacturer#2 k,BQms5UhoAF1B2Asi,fLib 33-964-337-5038
efully express deposits kindle after the deposits. final
8655.99 Supplier#000006330 RUSSIA 193810.00 Manufacturer#2 UozlaENr0ytKe2w6CeIEWFWn iO3S8Rae7Ou 32-561-198-3705
symptotes use about the express dolphins. requests use after the express platelets. final, ex
8638.36 Supplier#000002920 RUSSIA 75398.00 Manufacturer#1 Je2a8bszf3L 32-122-621-7549
ly quickly ironic requests. even requests whithout t
8638.36 Supplier#000002920 RUSSIA 170402.00 Manufacturer#3 Je2a8bszf3L 32-122-621-7549
ly quickly ironic requests. even requests whithout t
8607.69 Supplier#000006003 UNITED KINGDOM 76002.00 Manufacturer#2 EH9wADcEiuenM0NR08zDwMidw,52Y2RyILEiA 33-416-807-5206
ar, pending accounts. pending depende
8569.52 Supplier#000005936 RUSSIA 5935.00 Manufacturer#5 jXaNZ6vwnEWJ2ksLZJpjtgt0bY2a3AU 32-644-251-7916
. regular foxes nag carefully atop the regular, silent deposits. quickly regular packages
8564.12 Supplier#000000033 GERMANY 110032.00 Manufacturer#1

gfeKpYw3400L0SDywXA6Ya1Qmq1w6YB9f3R 17-138-897-9374
n sauternes along the regular asymptotes are regularly along the
8553.82 Supplier#000003979 ROMANIA 143978.00 Manufacturer#4
BfmVhCAnCMY3jzpjUMy4CNWs9 HzpdQR7INJU 29-124-646-4897
ic requests wake against the blithely unusual accounts. fluffily r
8517.23 Supplier#000009529 RUSSIA 37025.00 Manufacturer#5
e44R8o7JAIS9iMcr 32-565-297-8775
ove the even courts. furiously special platelets
8517.23 Supplier#000009529 RUSSIA 59528.00 Manufacturer#2
e44R8o7JAIS9iMcr 32-565-297-8775
ove the even courts. furiously special platelets
8503.70 Supplier#000006830 RUSSIA 44325.00 Manufacturer#4
BC4WFCYRUZyaIgchU 4S 32-147-878-5069
pades cajole. furious packages among the carefully express excuses boost furiously across th
8457.09 Supplier#000009456 UNITED KINGDOM 19455.00 Manufacturer#1
7SBhZs8gP1cJjT0Qf433YBk 33-858-440-4349
cing requests along the furiously unusual deposits promise among the furiously unus
8441.40 Supplier#000003817 FRANCE 141302.00 Manufacturer#2
hU3fz3xL78 16-339-356-5115
ely even ideas. ideas wake slyly furiously unusual instructions. pinto beans sleep ag
8432.89 Supplier#000003990 RUSSIA 191470.00 Manufacturer#1
wehBBp1RQbfxAYDASS75MsywmsKHRVdkrvNe6m 32-839-509-9301
ep furiously. packages should have to haggle slyly across the deposits. furiously regu
8431.40 Supplier#000002675 ROMANIA 5174.00 Manufacturer#1
HJFStOu9R5NGPOegKhgbzBdyvrG2yh8w 29-474-643-1443
ithely express pinto beans. blithely even foxes haggle. furiously regular theodol
8407.04 Supplier#000005406 RUSSIA 162889.00 Manufacturer#4
j7 gYF5RW8DC5UrjKC 32-626-152-4621
r the blithely regular packages. slyly ironic theodoli
8386.08 Supplier#000008518 FRANCE 36014.00 Manufacturer#3
2jqzqqAVe9crMVGP,n9nTsQXulNLTUYoJjEDcqWV 16-618-780-7481
blithely bold pains are carefully platelets. finally regular pinto beans sleep carefully special
8376.52 Supplier#000005306 UNITED KINGDOM 190267.00 Manufacturer#5
9t8Y8 QqSIsoADPt6NLdk,TP5zyRx41oBUlgoGc9 33-632-514-7931
ly final accounts sleep special, regular requests. furiously regular
8348.74 Supplier#000008851 FRANCE 66344.00 Manufacturer#4
nWxi7GwEbjhw1 16-796-240-2472
boldly final deposits. regular, even instructions detect slyly. fluffily unusual pinto bea
8338.58 Supplier#000007269 FRANCE 17268.00 Manufacturer#4
ZwhJSwABUoiB04,3 16-267-277-4365
iously final accounts. even pinto beans cajole slyly regular
8328.46 Supplier#000001744 ROMANIA 69237.00 Manufacturer#5
oLo3fV64q2,FKHa3p,qHnS7Yzv,ps8 29-330-728-5873
ep carefully-- even, careful packages are slyly along t
8307.93 Supplier#000003142 GERMANY 18139.00 Manufacturer#1
dqblvV8dCNAorGlJ 17-595-447-6026
olites wake furiously regular decoys. final requests nod
8231.61 Supplier#000009558 RUSSIA 192000.00 Manufacturer#2
mcdgen,yT1iJDHDS5fV 32-762-137-5858
foxes according to the furi
8152.61 Supplier#000002731 ROMANIA 15227.00 Manufacturer#4
nluXJCuY1tu 29-805-463-2030
special requests. even, regular warhorses affix among the final gr
8109.09 Supplier#000009186 FRANCE 99185.00 Manufacturer#1
wgfosrVPexl9pEXWywaqlBMDYYf 16-668-570-1402
tions haggle slyly about the sil
8102.62 Supplier#000003347 UNITED KINGDOM 18344.00 Manufacturer#5
m CtXS2S16i 33-454-274-8532
egrate with the slyly bold instructions. special foxes haggle silently among the
8046.07 Supplier#000008780 FRANCE 191222.00 Manufacturer#3
AczzuE0UK9osj ,Lx0Jmh 16-473-215-6395
onic platelets cajole after the regular instructions. permanently bold excuses
8042.09 Supplier#000003245 RUSSIA 135705.00 Manufacturer#4
Dh8Ikg39onrbOL4DyTfGw8a9oKUX3d9Y 32-836-132-8872

osits. packages cajole slyly. furiously regular deposits cajole slyly. q
8042.09 Supplier#000003245 RUSSIA 150729.00 Manufacturer#1 Dh8Ikg39onrbOL4DyTfGw8a9oKUX3d9Y 32-836-132-8872
osits. packages cajole slyly. furiously regular deposits cajole slyly. q
7992.40 Supplier#000006108 FRANCE 118574.00 Manufacturer#1 8tBydnTDwUqfBfFV4l3 16-974-998-8937
ironic ideas? fluffily even instructions wake. blithel
7980.65 Supplier#000001288 FRANCE 13784.00 Manufacturer#4 zE,7HgVPrCn 16-646-464-8247
ully bold courts. escapades nag slyly. furiously fluffy theodo
7950.37 Supplier#000008101 GERMANY 33094.00 Manufacturer#5 kkYvL6IuvojJgTNG IKkaXQDYgx8ILohj 17-627-663-8014
arefully unusual requests x-ray above the quickly final deposits.
7937.93 Supplier#000009012 ROMANIA 83995.00 Manufacturer#2 iUiTziH,Ek3i4lwSgunXMgrcTzwdb 29-250-925-9690
to the blithely ironic deposits nag sly
7914.45 Supplier#000001013 RUSSIA 125988.00 Manufacturer#2 riRcntps4KEDtYScjpMIWeYF6mNnR 32-194-698-3365
busily bold packages are dolphi
7912.91 Supplier#000004211 GERMANY 159180.00 Manufacturer#5 2wQRVovHrm3,v03IKzfTd,1PYsFXQFFOG 17-266-947-7315
ay furiously regular platelets. cou
7912.91 Supplier#000004211 GERMANY 184210.00 Manufacturer#4 2wQRVovHrm3,v03IKzfTd,1PYsFXQFFOG 17-266-947-7315
ay furiously regular platelets. cou
7894.56 Supplier#000007981 GERMANY 85472.00 Manufacturer#4 NSJ96vMROAbeXP 17-963-404-3760
ic platelets affix after the furiously
7887.08 Supplier#000009792 GERMANY 164759.00 Manufacturer#3 Y28ITVeYriT3kIGdV2K8fSZ V2UqT5H1Otz 17-988-938-4296
ckly around the carefully fluffy theodolites. slyly ironic pack
7871.50 Supplier#000007206 RUSSIA 104695.00 Manufacturer#1 3w fNCnrVmvJjE95sgWZzvW 32-432-452-7731
ironic requests. furiously final theodolites cajole. final, express packages sleep. quickly reg
7852.45 Supplier#000005864 RUSSIA 8363.00 Manufacturer#4 WCNfBPZeSXh3h,c 32-454-883-3821
usly unusual pinto beans. brave ideas sleep carefully quickly ironi
7850.66 Supplier#000001518 UNITED KINGDOM 86501.00 Manufacturer#1 ONda3YJiHKJOC 33-730-383-3892
ifts haggle fluffily pending pai
7843.52 Supplier#000006683 FRANCE 11680.00 Manufacturer#4 2Z0JGkiv01Y00oCFwUGfviIbhzCdy 16-464-517-8943
express, final pinto beans x-ray slyly asymptotes. unusual, unusual

100 rows processed.
Query Processed in 1.14 seconds.

-- Q3

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

L_ORDERKEY REVENUE O_ORDERDATE O_SHIPPRIORITY
2456423.00 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 4.35 seconds.
```

-- Q4

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

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


5 rows processed.
Query Processed in 0.36 seconds.
```

-- Q5

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

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


5 rows processed.
Query Processed in 3.48 seconds.
```

-- Q6

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

```
REVENUE
123141078.23


1 row processed.
```

Query Processed in 0.37 seconds.

-- Q7

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

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

4 rows processed.
```

Query Processed in 2.24 seconds.

-- Q8

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

```
O_YEAR          MKT_SHARE
1995.00         0.03
1996.00         0.04

2 rows processed.
```

Query Processed in 2.39 seconds.

-- Q9

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

NATION	O_YEAR	SUM_PROFIT
ALGERIA	1998.00	31342867.23
ALGERIA	1997.00	57138193.02
ALGERIA	1996.00	56140140.13
ALGERIA	1995.00	53051469.65
ALGERIA	1994.00	53867582.13
ALGERIA	1993.00	54942718.13
ALGERIA	1992.00	54628034.71
ARGENTINA	1998.00	30211185.71
ARGENTINA	1997.00	50805741.75
ARGENTINA	1996.00	51923746.58
ARGENTINA	1995.00	49298625.77
ARGENTINA	1994.00	50835610.11
ARGENTINA	1993.00	51646079.18
ARGENTINA	1992.00	50410314.99
BRAZIL	1998.00	27217924.38
BRAZIL	1997.00	48378669.20
BRAZIL	1996.00	50482870.36
BRAZIL	1995.00	47623383.63
BRAZIL	1994.00	47840165.73
BRAZIL	1993.00	49054694.04
BRAZIL	1992.00	48667639.08
CANADA	1998.00	30379833.77
CANADA	1997.00	50465052.31
CANADA	1996.00	52560501.39
CANADA	1995.00	52375332.81
CANADA	1994.00	52600364.66
CANADA	1993.00	52644504.07
CANADA	1992.00	53932871.70
CHINA	1998.00	31075466.16
CHINA	1997.00	50551874.45
CHINA	1996.00	51039293.88
CHINA	1995.00	49287534.62
CHINA	1994.00	50851090.07
CHINA	1993.00	54229629.83
CHINA	1992.00	52400529.37
EGYPT	1998.00	29054433.39
EGYPT	1997.00	50627611.45
EGYPT	1996.00	49542212.84
EGYPT	1995.00	48311550.32
EGYPT	1994.00	49790644.74
EGYPT	1993.00	48904292.97
EGYPT	1992.00	49434932.62
ETHIOPIA	1998.00	28040717.27
ETHIOPIA	1997.00	47455009.87
ETHIOPIA	1996.00	46491097.57
ETHIOPIA	1995.00	46804449.30
ETHIOPIA	1994.00	48516143.92
ETHIOPIA	1993.00	46551891.56
ETHIOPIA	1992.00	44934648.64
FRANCE	1998.00	32226407.84
FRANCE	1997.00	47121485.86
FRANCE	1996.00	47263135.50
FRANCE	1995.00	47275997.57
FRANCE	1994.00	47067209.33
FRANCE	1993.00	51163370.11
FRANCE	1992.00	47846235.33
GERMANY	1998.00	28624942.66
GERMANY	1997.00	49309074.88
GERMANY	1996.00	49918683.17
GERMANY	1995.00	52650718.72
GERMANY	1994.00	50346900.42
GERMANY	1993.00	50991895.81
GERMANY	1992.00	48274126.10
INDIA	1998.00	29943144.35
INDIA	1997.00	50665453.23
INDIA	1996.00	50283092.29
INDIA	1995.00	50006774.64
INDIA	1994.00	48995190.76
INDIA	1993.00	50286902.85
INDIA	1992.00	50850329.40
INDONESIA	1998.00	27672340.00
INDONESIA	1997.00	50512145.73
INDONESIA	1996.00	51653060.12

INDONESIA	1995.00	51508779.59
INDONESIA	1994.00	52817950.32
INDONESIA	1993.00	47959994.96
INDONESIA	1992.00	51776605.03
IRAN	1998.00	29065736.24
IRAN	1997.00	50042063.05
IRAN	1996.00	50926653.19
IRAN	1995.00	51249667.65
IRAN	1994.00	50337085.87
IRAN	1993.00	51730763.49
IRAN	1992.00	49955856.56
IRAQ	1998.00	31624551.00
IRAQ	1997.00	55121749.02
IRAQ	1996.00	55897663.79
IRAQ	1995.00	54815472.52
IRAQ	1994.00	54408516.13
IRAQ	1993.00	53633167.98
IRAQ	1992.00	55891939.34
JAPAN	1998.00	27934179.67
JAPAN	1997.00	44517162.55
JAPAN	1996.00	42545606.12
JAPAN	1995.00	43749356.40
JAPAN	1994.00	44840243.07
JAPAN	1993.00	44660015.53
JAPAN	1992.00	45410249.12
JORDAN	1998.00	26901488.58
JORDAN	1997.00	45471878.41
JORDAN	1996.00	46794325.79
JORDAN	1995.00	45178828.58
JORDAN	1994.00	45333636.51
JORDAN	1993.00	47971496.10
JORDAN	1992.00	44717239.18
KENYA	1998.00	28597614.34
KENYA	1997.00	47949733.73
KENYA	1996.00	46886924.62
KENYA	1995.00	46072338.76
KENYA	1994.00	45772061.17
KENYA	1993.00	46308728.23
KENYA	1992.00	47257780.84
MOROCCO	1998.00	26732115.58
MOROCCO	1997.00	45637304.25
MOROCCO	1996.00	45558221.75
MOROCCO	1995.00	47851318.89
MOROCCO	1994.00	46272172.94
MOROCCO	1993.00	46764326.18
MOROCCO	1992.00	48122783.58
MOZAMBIQUE	1998.00	30712392.01
MOZAMBIQUE	1997.00	50316528.76
MOZAMBIQUE	1996.00	51640320.25
MOZAMBIQUE	1995.00	50693774.51
MOZAMBIQUE	1994.00	49253277.63
MOZAMBIQUE	1993.00	49153016.54
MOZAMBIQUE	1992.00	48247551.85
PERU	1998.00	29326102.32
PERU	1997.00	49753780.40
PERU	1996.00	50935170.29
PERU	1995.00	53309883.41
PERU	1994.00	50643531.80
PERU	1993.00	51584622.00
PERU	1992.00	47523899.05
ROMANIA	1998.00	30368667.40
ROMANIA	1997.00	50365683.85
ROMANIA	1996.00	49598999.01
ROMANIA	1995.00	47537642.87
ROMANIA	1994.00	51455283.01
ROMANIA	1993.00	50407136.89
ROMANIA	1992.00	48185385.13
RUSSIA	1998.00	28322384.03
RUSSIA	1997.00	50106685.18
RUSSIA	1996.00	51753342.43
RUSSIA	1995.00	49215820.36
RUSSIA	1994.00	52205666.44
RUSSIA	1993.00	51860230.03
RUSSIA	1992.00	53251677.15
SAUDI ARABIA	1998.00	31541259.81
SAUDI ARABIA	1997.00	52438750.81
SAUDI ARABIA	1996.00	52543737.82
SAUDI ARABIA	1995.00	52938696.53
SAUDI ARABIA	1994.00	51389601.97
SAUDI ARABIA	1993.00	52937508.88
SAUDI ARABIA	1992.00	54843459.64
UNITED KINGDOM	1998.00	28494874.00
UNITED KINGDOM	1997.00	49381810.90
UNITED KINGDOM	1996.00	51386853.96
UNITED KINGDOM	1995.00	51509586.79
UNITED KINGDOM	1994.00	48086499.71
UNITED KINGDOM	1993.00	49166827.22
UNITED KINGDOM	1992.00	49349122.08
UNITED STATES	1998.00	25126238.95
UNITED STATES	1997.00	50077306.42
UNITED STATES	1996.00	48048649.47
UNITED STATES	1995.00	48809032.42
UNITED STATES	1994.00	49296747.18
UNITED STATES	1993.00	48029946.80
UNITED STATES	1992.00	48671944.50
VIETNAM	1998.00	30442736.06
VIETNAM	1997.00	50309179.79
VIETNAM	1996.00	50488161.41
VIETNAM	1995.00	49658284.61
VIETNAM	1994.00	50596057.26
VIETNAM	1993.00	50953919.15
VIETNAM	1992.00	49613838.32

175 rows processed.

Query Processed in 8.96 seconds.

-- Q10

```
select * from (
select
c_custkey,
c_name,
sum(l_extendedprice * (1 - l_discount)) as revenue,
c_acctbal,
n_name,
c_address,
c_phone,
c_comment
from
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
sits. slyly regular requests sleep alongside of the regular inst
143347.00 Customer#000143347 721002.69
2557.47 EGYPT
1aReFYv,Kw4 14-742-935-3718
ggle carefully enticing requests. final deposits use bold, bold pinto beans. ironic, idle re
60838.00 Customer#000060838 679127.31
2454.77 BRAZIL
64EaJ5vMAHWJlBOxJklpNc2RJiWE 12-913-494-9813
 need to boost against the slyly regular account
101998.00 Customer#000101998 637029.57
3790.89 UNITED KINGDOM
01c9CILnNtfOQYmZj 33-593-865-6378
ress foxes wake slyly after the bold excuses. ironic platelets are furiously carefully bold theodolites
125341.00 Customer#000125341 633508.09
4983.51 GERMANY
S29ODD6bceU8QSuuEJznkNaK 17-582-695-5962
arefully even depths. blithely even excuses sleep furiously. foxes use except the dependencies. ca
25501.00 Customer#000025501 620269.78
7725.04 ETHIOPIA
 W556MXuoiaYCCZamJI,Rn0B4ACUGdkQ8DZ 15-874-808-6793
he pending instructions wake carefully at the pinto beans. regular, final instructions along the slyly fina
115831.00 Customer#000115831 596423.87
5098.10 FRANCE
rFeBbEEyk dl ne7zV5fDrmiq1oK09wV7pxqCgIc 16-715-386-3788
l somas sleep. furiously final deposits wake blithely regular pinto b
84223.00 Customer#000084223 594998.02
528.65 UNITED KINGDOM
nAVZCs6BaWap rrM27N 2qBnzc5WBauxbA 33-442-824-8191
 slyly final deposits haggle regular, pending dependencies. pending escapades wake
54289.00 Customer#000054289 585603.39
5583.02 IRAN
vXCxoCsU0Bad5JQI ,oobkZ 20-834-292-4707
ely special foxes are quickly finally ironic p
39922.00 Customer#000039922 584878.11
7321.11 GERMANY
Zgy4s50l2GKN4pLDPBU8m342gIw6R 17-147-757-8036
y final requests. furiously final foxes cajole blithely special platelets. f
6226.00 Customer#000006226 576783.76
2230.09 UNITED KINGDOM
8gPu8,NPGkfyQQ0hcIYUGPIBWc,ybP5g, 33-657-701-3391
ending platelets along the express deposits cajole carefully final
922.00 Customer#000000922 576767.53
3869.25 GERMANY
Az9RFaut7NkPnc5zSD2PwHgVwr4jRzq 17-945-916-9648
luffily fluffy deposits. packages c
147946.00 Customer#000147946 576455.13
2030.13 ALGERIA
iANyZHjqhyy7Ajah0pTrYyhJ 10-886-956-3143
```

ithely ironic deposits haggle blithely ironic requests.
quickly regu
115640.00 Customer#000115640 569341.19
6436.10 ARGENTINA
Vtgfia9qI 7EpHgecU1X 11-411-543-4901
ost slyly along the patterns; pinto be
73606.00 Customer#000073606 568656.86
1785.67 JAPAN
xuR0Tro5yChDfOCrjkd2ol 22-437-653-6966
he furiously regular ideas. slowly
110246.00 Customer#000110246 566842.98
7763.35 VIETNAM
7KzflgX MDOq7sOkI 31-943-426-9837
egular deposits serve blithely above the fl
142549.00 Customer#000142549 563537.24
5085.99 INDONESIA
ChqEoK43OysjdHbtKCp6dKqjNyvvi9 19-955-
562-2398
sleep pending courts. ironic deposits against the carefully
unusual platelets cajole carefully express accounts.
146149.00 Customer#000146149 557254.99
1791.55 ROMANIA
s87fvzFQpU 29-744-164-6487
of the slyly silent accounts. quickly final accounts across
the
52528.00 Customer#000052528 556397.35
551.79 ARGENTINA
NFztyTOR10UOJ 11-208-192-3205
deposits hinder. blithely pending asymptotes breach slyly
regular re
23431.00 Customer#000023431 554269.54
3381.86 ROMANIA
HgiV0phqhaIa9aydNoIlb 29-915-458-2654
nusual, even instructions: furiously stealthy n

20 rows processed.
Query Processed in 1.35 seconds.

-- Q11

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

PS_PARTKEY	VALUE
129760.00	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

delete lines

56694.00 7936015.95
8114.00 7933921.88
71518.00 7932261.69
72922.00 7930400.64
146699.00 7929167.40
92387.00 7928972.67
186289.00 7928786.19
95952.00 7927972.78
196514.00 7927180.70
4403.00 7925729.04
2267.00 7925649.37
45924.00 7925047.68
11493.00 7916722.23
104478.00 7916253.60
166794.00 7913842.00
161995.00 7910874.27
23538.00 7909752.06
41093.00 7909579.92
112073.00 7908617.57
92814.00 7908262.50
88919.00 7907992.50
79753.00 7907933.88
108765.00 7905338.98
146530.00 7905336.60
71475.00 7903367.58
36289.00 7901946.50
61739.00 7900794.00
52338.00 7898638.08
194299.00 7898421.24
105235.00 7897829.94
77207.00 7897752.72
96712.00 7897575.27
10157.00 7897046.25
171154.00 7896814.50
79373.00 7896186.00
113808.00 7893353.88
27901.00 7892952.00
128820.00 7892882.72
25891.00 7890511.20
122819.00 7888881.02
154731.00 7888301.33
101674.00 7879324.60
51968.00 7879102.21
72073.00 7877736.11
5182.00 7874521.73

1048 rows processed.
Query Processed in 1.56 seconds.

-- Q12

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

```
order by
        l_shipmode

L_SHIPMODE HIGH_LINE_COUNT
LOW_LINE_COUNT
MAIL       6202.00             9324.00
SHIP       6200.00             9262.00


2 rows processed.
Query Processed in 1.63 seconds.


-- Q13


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

C_COUNT          CUSTDIST
0.00             50005.00
9.00             6641.00
10.00             6532.00
11.00             6014.00
8.00             5937.00
12.00             5639.00
13.00             5024.00
19.00             4793.00
7.00             4687.00
17.00             4587.00
18.00             4529.00
20.00             4516.00
15.00             4505.00
14.00             4446.00
16.00             4273.00
21.00             4190.00
22.00             3623.00
6.00             3265.00
23.00             3225.00
24.00             2742.00
25.00             2086.00
5.00             1948.00
26.00             1612.00
27.00             1179.00
4.00             1007.00
28.00             893.00
29.00             593.00
3.00             415.00
30.00             376.00
31.00             226.00
32.00             148.00
2.00             134.00
33.00             75.00
34.00             50.00
35.00             37.00
1.00             17.00
36.00             14.00
38.00             5.00
37.00             5.00
40.00             4.00
41.00             2.00
39.00             1.00


42 rows processed.
Query Processed in 2.95 seconds.


-- Q14

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

PROMO_REVENUE
16.38


1 row processed.
Query Processed in 0.29 seconds.
```

```
-- Q15a


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

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


1 row processed.
Query Processed in 1.01 seconds.


-- Q16


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

P_BRAND    P_TYPE                 P_SIZE
SUPPLIER_CNT
Brand#41   MEDIUM BRUSHED TIN       3.00
28.00
Brand#54   STANDARD BRUSHED COPPER   14.00
27.00
Brand#11   STANDARD BRUSHED TIN      23.00
24.00
Brand#11   STANDARD BURNISHED BRASS  36.00
24.00
Brand#15   MEDIUM ANODIZED NICKEL    3.00
24.00
Brand#15   SMALL ANODIZED BRASS      45.00
24.00
Brand#15   SMALL BURNISHED NICKEL    19.00
24.00
Brand#21   MEDIUM ANODIZED COPPER    3.00
24.00
Brand#22   SMALL BRUSHED NICKEL      3.00
24.00
Brand#22   SMALL BURNISHED BRASS     19.00
24.00
Brand#25   MEDIUM BURNISHED COPPER   36.00
24.00
Brand#31   PROMO POLISHED COPPER     36.00
24.00
Brand#33   LARGE POLISHED TIN        23.00
24.00
Brand#33   PROMO POLISHED STEEL      14.00
24.00
Brand#35   PROMO BRUSHED NICKEL      14.00
24.00
Brand#41   ECONOMY BRUSHED STEEL     9.00
24.00
```

Brand#41 ECONOMY POLISHED TIN 19.00 24.00
Brand#41 LARGE PLATED COPPER 36.00 24.00
Brand#42 ECONOMY PLATED BRASS 3.00 24.00
Brand#42 STANDARD POLISHED TIN 49.00 24.00
Brand#43 PROMO BRUSHED TIN 3.00 24.00
Brand#43 SMALL ANODIZED COPPER 36.00 24.00
Brand#44 STANDARD POLISHED NICKEL 3.00 24.00
Brand#52 ECONOMY PLATED TIN 14.00 24.00
Brand#52 STANDARD BURNISHED NICKEL 3.00 24.00
Brand#53 MEDIUM ANODIZED STEEL 14.00 24.00
Brand#14 PROMO ANODIZED NICKEL 45.00 23.00
Brand#32 ECONOMY PLATED BRASS 9.00 23.00
Brand#52 SMALL ANODIZED COPPER 3.00 23.00
Brand#11 ECONOMY BRUSHED COPPER 45.00 20.00
Brand#11 ECONOMY PLATED BRASS 23.00 20.00
Brand#11 LARGE BRUSHED COPPER 49.00 20.00
Brand#11 LARGE POLISHED COPPER 49.00 20.00
Brand#12 STANDARD ANODIZED TIN 49.00 20.00
Brand#12 STANDARD PLATED BRASS 19.00 20.00
Brand#13 ECONOMY BRUSHED BRASS 9.00 20.00
Brand#13 ECONOMY BURNISHED STEEL 14.00 20.00
Brand#13 LARGE BURNISHED NICKEL 19.00 20.00
Brand#13 MEDIUM BURNISHED COPPER 36.00 20.00
Brand#13 SMALL BRUSHED TIN 45.00 20.00
Brand#13 STANDARD ANODIZED COPPER 3.00 20.00
Brand#13 STANDARD PLATED NICKEL 23.00 20.00
Brand#14 ECONOMY ANODIZED COPPER 14.00 20.00
Brand#14 ECONOMY PLATED TIN 36.00 20.00
Brand#14 ECONOMY POLISHED NICKEL 3.00 20.00
Brand#14 MEDIUM ANODIZED NICKEL 3.00 20.00
Brand#14 SMALL POLISHED TIN 14.00 20.00
Brand#15 MEDIUM ANODIZED COPPER 9.00 20.00
Brand#15 MEDIUM PLATED TIN 23.00 20.00
Brand#15 PROMO PLATED BRASS 14.00 20.00
Brand#15 SMALL ANODIZED COPPER 45.00 20.00
Brand#15 SMALL PLATED COPPER 49.00 20.00
Brand#15 STANDARD PLATED TIN 3.00 20.00
Brand#21 LARGE ANODIZED COPPER 36.00 20.00
Brand#21 LARGE BRUSHED TIN 3.00 20.00
Brand#21 MEDIUM ANODIZED COPPER 14.00 20.00
Brand#21 PROMO BRUSHED TIN 36.00 20.00
Brand#21 PROMO POLISHED NICKEL 45.00 20.00
Brand#21 SMALL ANODIZED COPPER 9.00 20.00
Brand#21 SMALL POLISHED NICKEL 23.00 20.00
Brand#22 LARGE ANODIZED COPPER 36.00 20.00
Brand#22 LARGE BRUSHED COPPER 49.00 20.00
Brand#22 PROMO ANODIZED TIN 49.00 20.00
Brand#22 PROMO POLISHED BRASS 45.00 20.00
Brand#22 SMALL BURNISHED STEEL 45.00 20.00
Brand#23 MEDIUM ANODIZED STEEL 45.00 20.00
Brand#23 PROMO POLISHED STEEL 23.00 20.00
Brand#23 STANDARD BRUSHED TIN 14.00 20.00
Brand#23 STANDARD PLATED NICKEL 36.00 20.00
Brand#24 PROMO PLATED COPPER 49.00 20.00
Brand#24 PROMO PLATED STEEL 49.00 20.00
Brand#24 PROMO POLISHED STEEL 9.00 20.00

Brand#24 STANDARD BRUSHED TIN 36.00
20.00
Brand#25 LARGE ANODIZED BRASS 3.00
20.00
Brand#25 PROMO BURNISHED TIN 3.00
20.00
Brand#31 ECONOMY POLISHED NICKEL 3.00
20.00
Brand#31 MEDIUM PLATED TIN 45.00
20.00
Brand#31 SMALL ANODIZED STEEL 14.00
20.00
Brand#32 ECONOMY ANODIZED COPPER 36.00
20.00
Brand#32 ECONOMY BRUSHED NICKEL 49.00
20.00
Brand#32 LARGE ANODIZED TIN 19.00
20.00
Brand#32 MEDIUM BURNISHED COPPER 19.00
20.00
Brand#32 SMALL ANODIZED STEEL 45.00
20.00
Brand#33 ECONOMY POLISHED COPPER 19.00
20.00
Brand#33 PROMO PLATED NICKEL 14.00
20.00
Brand#33 SMALL POLISHED TIN 9.00
20.00
Brand#33 STANDARD ANODIZED BRASS 49.00
20.00
Brand#33 STANDARD BURNISHED BRASS 45.00
20.00
Brand#34 ECONOMY BRUSHED NICKEL 49.00
20.00
Brand#34 LARGE BRUSHED BRASS 19.00
20.00
Brand#34 SMALL BRUSHED TIN 3.00
20.00
Brand#34 STANDARD PLATED COPPER 9.00
20.00
Brand#35 LARGE ANODIZED NICKEL 3.00
20.00
Brand#35 MEDIUM ANODIZED BRASS 45.00
20.00
Brand#35 MEDIUM ANODIZED STEEL 23.00
20.00
Brand#35 PROMO ANODIZED COPPER 49.00
20.00
Brand#35 SMALL POLISHED COPPER 14.00
20.00
Brand#41 LARGE ANODIZED STEEL 3.00
20.00
Brand#41 LARGE BRUSHED NICKEL 23.00
20.00
Brand#41 LARGE BURNISHED COPPER 3.00
20.00
Brand#41 MEDIUM PLATED STEEL 19.00
20.00
Brand#41 SMALL BURNISHED COPPER 23.00
20.00
Brand#42 MEDIUM BURNISHED BRASS 14.00
20.00
Brand#42 SMALL BURNISHED COPPER 3.00
20.00
Brand#43 ECONOMY POLISHED COPPER 9.00
20.00
Brand#43 SMALL PLATED STEEL 3.00
20.00
Brand#43 STANDARD BURNISHED TIN 23.00
20.00
Brand#44 LARGE ANODIZED STEEL 23.00
20.00
Brand#44 PROMO ANODIZED TIN 23.00
20.00
Brand#51 ECONOMY BRUSHED BRASS 49.00
20.00
Brand#51 ECONOMY POLISHED NICKEL 9.00
20.00
Brand#51 MEDIUM BRUSHED TIN 9.00
20.00
Brand#51 MEDIUM PLATED BRASS 9.00
20.00
Brand#51 PROMO BURNISHED BRASS 9.00
20.00
Brand#51 SMALL PLATED NICKEL 49.00
20.00
Brand#51 STANDARD ANODIZED NICKEL 49.00
20.00
Brand#51 STANDARD BRUSHED COPPER 3.00
20.00
Brand#52 ECONOMY ANODIZED BRASS 3.00
20.00
Brand#52 ECONOMY BRUSHED COPPER 49.00
20.00
Brand#52 LARGE ANODIZED NICKEL 45.00
20.00
Brand#52 MEDIUM ANODIZED TIN 23.00
20.00
Brand#52 MEDIUM BURNISHED TIN 45.00
20.00
Brand#52 SMALL PLATED COPPER 36.00
20.00
Brand#52 STANDARD ANODIZED BRASS 45.00
20.00
Brand#53 ECONOMY PLATED COPPER 45.00
20.00
Brand#53 PROMO ANODIZED COPPER 49.00
20.00
Brand#53 PROMO BRUSHED COPPER 23.00
20.00
Brand#53 PROMO PLATED TIN 19.00
20.00

Brand#53 PROMO POLISHED NICKEL 3.00
20.00
Brand#53 SMALL ANODIZED STEEL 9.00
20.00
Brand#53 SMALL BRUSHED COPPER 3.00
20.00
Brand#53 SMALL BRUSHED NICKEL 3.00
20.00
Brand#54 ECONOMY PLATED STEEL 9.00
20.00
Brand#54 ECONOMY POLISHED TIN 3.00
20.00
Brand#54 SMALL BRUSHED BRASS 19.00
20.00
Brand#55 MEDIUM ANODIZED COPPER 3.00
20.00
Brand#55 PROMO BURNISHED STEEL 14.00
20.00
Brand#55 PROMO POLISHED NICKEL 49.00
20.00
Brand#55 STANDARD ANODIZED BRASS 19.00
20.00
Brand#55 STANDARD BURNISHED COPPER 45.00
20.00
Brand#43 ECONOMY ANODIZED TIN 3.00
19.00
Brand#11 ECONOMY ANODIZED BRASS 14.00
16.00
Brand#11 ECONOMY ANODIZED BRASS 23.00
16.00
Brand#11 ECONOMY ANODIZED COPPER 14.00
16.00
Brand#11 ECONOMY BRUSHED BRASS 49.00
16.00
Brand#11 ECONOMY BRUSHED STEEL 19.00
16.00
Brand#11 ECONOMY BURNISHED NICKEL 23.00
16.00
Brand#11 LARGE ANODIZED COPPER 14.00
16.00
Brand#11 LARGE BRUSHED TIN 45.00
16.00
Brand#11 LARGE BURNISHED COPPER 23.00
16.00
Brand#11 LARGE BURNISHED NICKEL 36.00
16.00
Brand#11 LARGE PLATED STEEL 14.00
16.00
Brand#11 MEDIUM BRUSHED NICKEL 14.00
16.00
Brand#11 MEDIUM BRUSHED STEEL 49.00
16.00
Brand#11 MEDIUM BURNISHED NICKEL 49.00
16.00
Brand#11 MEDIUM BURNISHED TIN 3.00
16.00
Brand#11 MEDIUM PLATED COPPER 9.00
16.00
Brand#11 PROMO ANODIZED BRASS 19.00
16.00
Brand#11 PROMO ANODIZED BRASS 49.00
16.00
Brand#11 PROMO ANODIZED STEEL 45.00
16.00
Brand#11 PROMO PLATED BRASS 45.00
16.00
delete lines
Brand#55 PROMO POLISHED COPPER 3.00
4.00
Brand#55 PROMO POLISHED COPPER 19.00
4.00
Brand#55 PROMO POLISHED COPPER 45.00
4.00
Brand#55 PROMO POLISHED COPPER 49.00
4.00
Brand#55 PROMO POLISHED NICKEL 3.00
4.00
Brand#55 PROMO POLISHED NICKEL 14.00
4.00
Brand#55 PROMO POLISHED NICKEL 19.00
4.00
Brand#55 PROMO POLISHED NICKEL 23.00
4.00
Brand#55 PROMO POLISHED NICKEL 36.00
4.00
Brand#55 PROMO POLISHED STEEL 19.00
4.00
Brand#55 PROMO POLISHED STEEL 45.00
4.00
Brand#55 PROMO POLISHED STEEL 49.00
4.00
Brand#55 PROMO POLISHED TIN 3.00
4.00
Brand#55 PROMO POLISHED TIN 9.00
4.00
Brand#55 PROMO POLISHED TIN 14.00
4.00
Brand#55 PROMO POLISHED TIN 19.00
4.00
Brand#55 PROMO POLISHED TIN 23.00
4.00
Brand#55 PROMO POLISHED TIN 36.00
4.00
Brand#55 PROMO POLISHED TIN 45.00
4.00
Brand#55 PROMO POLISHED TIN 49.00
4.00
Brand#55 SMALL ANODIZED BRASS 23.00
4.00
Brand#55 SMALL ANODIZED BRASS 36.00
4.00
Brand#55 SMALL ANODIZED BRASS 45.00
4.00

Brand#55 SMALL ANODIZED COPPER 9.00 4.00
Brand#55 SMALL ANODIZED COPPER 19.00 4.00
Brand#55 SMALL ANODIZED COPPER 23.00 4.00
Brand#55 SMALL ANODIZED NICKEL 9.00 4.00
Brand#55 SMALL ANODIZED NICKEL 14.00 4.00
Brand#55 SMALL ANODIZED NICKEL 23.00 4.00
Brand#55 SMALL ANODIZED NICKEL 36.00 4.00
Brand#55 SMALL ANODIZED NICKEL 45.00 4.00
Brand#55 SMALL ANODIZED STEEL 36.00 4.00
Brand#55 SMALL ANODIZED TIN 9.00 4.00
Brand#55 SMALL ANODIZED TIN 36.00 4.00
Brand#55 SMALL ANODIZED TIN 45.00 4.00
Brand#55 SMALL ANODIZED TIN 49.00 4.00
Brand#55 SMALL BRUSHED BRASS 9.00 4.00
Brand#55 SMALL BRUSHED BRASS 36.00 4.00
Brand#55 SMALL BRUSHED COPPER 3.00 4.00
Brand#55 SMALL BRUSHED COPPER 9.00 4.00
Brand#55 SMALL BRUSHED COPPER 19.00 4.00
Brand#55 SMALL BRUSHED COPPER 23.00 4.00
Brand#55 SMALL BRUSHED NICKEL 3.00 4.00
Brand#55 SMALL BRUSHED NICKEL 9.00 4.00
Brand#55 SMALL BRUSHED NICKEL 19.00 4.00
Brand#55 SMALL BRUSHED NICKEL 23.00 4.00
Brand#55 SMALL BRUSHED NICKEL 45.00 4.00
Brand#55 SMALL BRUSHED NICKEL 49.00 4.00
Brand#55 SMALL BRUSHED STEEL 3.00 4.00
Brand#55 SMALL BRUSHED STEEL 14.00 4.00
Brand#55 SMALL BRUSHED STEEL 19.00 4.00
Brand#55 SMALL BRUSHED STEEL 23.00 4.00
Brand#55 SMALL BRUSHED STEEL 45.00 4.00
Brand#55 SMALL BRUSHED STEEL 49.00 4.00
Brand#55 SMALL BRUSHED TIN 9.00 4.00
Brand#55 SMALL BRUSHED TIN 49.00 4.00
Brand#55 SMALL BURNISHED BRASS 14.00 4.00
Brand#55 SMALL BURNISHED BRASS 23.00 4.00
Brand#55 SMALL BURNISHED COPPER 3.00 4.00
Brand#55 SMALL BURNISHED COPPER 9.00 4.00
Brand#55 SMALL BURNISHED COPPER 36.00 4.00
Brand#55 SMALL BURNISHED NICKEL 9.00 4.00
Brand#55 SMALL BURNISHED NICKEL 19.00 4.00
Brand#55 SMALL BURNISHED NICKEL 36.00 4.00
Brand#55 SMALL BURNISHED NICKEL 45.00 4.00
Brand#55 SMALL BURNISHED STEEL 14.00 4.00
Brand#55 SMALL BURNISHED TIN 9.00 4.00
Brand#55 SMALL BURNISHED TIN 23.00 4.00
Brand#55 SMALL PLATED COPPER 3.00 4.00
Brand#55 SMALL PLATED COPPER 14.00 4.00
Brand#55 SMALL PLATED COPPER 36.00 4.00
Brand#55 SMALL PLATED COPPER 49.00 4.00
Brand#55 SMALL PLATED NICKEL 14.00 4.00
Brand#55 SMALL PLATED NICKEL 49.00 4.00
Brand#55 SMALL PLATED STEEL 3.00 4.00
Brand#55 SMALL PLATED STEEL 23.00 4.00
Brand#55 SMALL PLATED STEEL 36.00 4.00
Brand#55 SMALL PLATED TIN 36.00 4.00
Brand#55 SMALL PLATED TIN 45.00 4.00

Brand#55 SMALL POLISHED BRASS 9.00 4.00
Brand#55 SMALL POLISHED BRASS 19.00 4.00
Brand#55 SMALL POLISHED BRASS 49.00 4.00
Brand#55 SMALL POLISHED COPPER 19.00 4.00
Brand#55 SMALL POLISHED COPPER 23.00 4.00
Brand#55 SMALL POLISHED COPPER 36.00 4.00
Brand#55 SMALL POLISHED COPPER 45.00 4.00
Brand#55 SMALL POLISHED COPPER 49.00 4.00
Brand#55 SMALL POLISHED NICKEL 9.00 4.00
Brand#55 SMALL POLISHED NICKEL 14.00 4.00
Brand#55 SMALL POLISHED NICKEL 19.00 4.00
Brand#55 SMALL POLISHED NICKEL 23.00 4.00
Brand#55 SMALL POLISHED NICKEL 45.00 4.00
Brand#55 SMALL POLISHED NICKEL 49.00 4.00
Brand#55 SMALL POLISHED STEEL 19.00 4.00
Brand#55 SMALL POLISHED STEEL 45.00 4.00
Brand#55 SMALL POLISHED TIN 14.00 4.00
Brand#55 SMALL POLISHED TIN 23.00 4.00
Brand#55 SMALL POLISHED TIN 45.00 4.00
Brand#55 STANDARD ANODIZED BRASS 9.00 4.00
Brand#55 STANDARD ANODIZED BRASS 23.00 4.00
Brand#55 STANDARD ANODIZED BRASS 49.00 4.00
Brand#55 STANDARD ANODIZED COPPER 9.00 4.00
Brand#55 STANDARD ANODIZED COPPER 14.00 4.00
Brand#55 STANDARD ANODIZED COPPER 45.00 4.00
Brand#55 STANDARD ANODIZED NICKEL 3.00 4.00
Brand#55 STANDARD ANODIZED NICKEL 14.00 4.00
Brand#55 STANDARD ANODIZED NICKEL 45.00 4.00
Brand#55 STANDARD ANODIZED NICKEL 49.00 4.00
Brand#55 STANDARD ANODIZED STEEL 3.00 4.00
Brand#55 STANDARD ANODIZED STEEL 14.00 4.00
Brand#55 STANDARD ANODIZED TIN 14.00 4.00
Brand#55 STANDARD ANODIZED TIN 36.00 4.00
Brand#55 STANDARD ANODIZED TIN 45.00 4.00
Brand#55 STANDARD BRUSHED BRASS 9.00 4.00
Brand#55 STANDARD BRUSHED BRASS 19.00 4.00
Brand#55 STANDARD BRUSHED COPPER 14.00 4.00
Brand#55 STANDARD BRUSHED COPPER 19.00 4.00
Brand#55 STANDARD BRUSHED NICKEL 3.00 4.00
Brand#55 STANDARD BRUSHED NICKEL 36.00 4.00
Brand#55 STANDARD BRUSHED STEEL 9.00 4.00
Brand#55 STANDARD BRUSHED STEEL 14.00 4.00
Brand#55 STANDARD BRUSHED STEEL 19.00 4.00
Brand#55 STANDARD BRUSHED STEEL 49.00 4.00
Brand#55 STANDARD BRUSHED TIN 19.00 4.00
Brand#55 STANDARD BRUSHED TIN 49.00 4.00
Brand#55 STANDARD BURNISHED BRASS 9.00 4.00
Brand#55 STANDARD BURNISHED BRASS 19.00 4.00
Brand#55 STANDARD BURNISHED BRASS 23.00 4.00
Brand#55 STANDARD BURNISHED BRASS 36.00 4.00
Brand#55 STANDARD BURNISHED COPPER 3.00 4.00
Brand#55 STANDARD BURNISHED NICKEL 9.00 4.00
Brand#55 STANDARD BURNISHED NICKEL 49.00 4.00
Brand#55 STANDARD BURNISHED STEEL 19.00 4.00
Brand#55 STANDARD BURNISHED STEEL 23.00 4.00
Brand#55 STANDARD BURNISHED STEEL 36.00 4.00

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

18314 rows processed.
Query Processed in 1.38 seconds.

-- Q17

```
select
sum(l_extendedprice) / 7.0 as avg_yearly
from
```

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

AVG_YEARLY
348406.05


1 row processed.
Query Processed in 1.70 seconds.


-- Q18


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

C_NAME                C_CUSTKEY
O_ORDERKEY            O_ORDERDATE
O_TOTALPRICE          SUM(L_QUANTITY)
Customer#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      REVENUE
1998-03-15
399195.47       301.00
Customer#000082441      82441.00        857959.00
1994-02-07
382579.74       305.00
Customer#000088703      88703.00        2995076.00
1994-01-30
363812.12       302.00


57 rows processed.
Query Processed in 5.62 seconds.


-- Q19


select
sum(l_extendedprice* (1 - l_discount)) as revenue
from
lineitem,
part
where
(
p_partkey = l_partkey
and p_brand = 'Brand#12'
and p_container in ('SM CASE', 'SM BOX', 'SM PACK',
'SM PKG')
and l_quantity >= 1 and l_quantity <= 1 + 10
and p_size between 1 and 5
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
)
or
(
p_partkey = l_partkey
and p_brand = 'Brand#23'
and p_container in ('MED BAG', 'MED BOX', 'MED
PKG', 'MED PACK')
and l_quantity >= 10 and l_quantity <= 10 + 10
and p_size between 1 and 10
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
)
or
(
p_partkey = l_partkey
and p_brand = 'Brand#34'
and p_container in ('LG CASE', 'LG BOX', 'LG PACK',
'LG PKG')
and l_quantity >= 20 and l_quantity <= 20 + 10
and p_size between 1 and 15
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
)
REVENUE
3083843.06


1 row processed.
Query Processed in 2.48 seconds.


-- Q20


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


S_NAME                  S_ADDRESS
Supplier#000000020
iybAE,RmTymrZVYaFZva2SH,j
Supplier#000000091
YV45D7TkfdQanOOZ7q9QxkyGUapU1oOWU6q3
```

Supplier#000000197
YC2Acon6kjY3zj3Fbxs2k4Vdf7X0cd2F
Supplier#000000226 83qOdU2EYRdPQAQhEtn
GRZEd
Supplier#000000285
Br7e1nnt1yxrw6ImgpJ7YdhFDjuBf
Supplier#000000378 FfbhyCxWvcPrO8ltp9
Supplier#000000402
i9Sw4DoyMhzhKXCH9By,AYSgmD
Supplier#000000530 0qwCMwobKY
OcmLyfRXlagA8ukENJv,
Supplier#000000688 D
fw5ocppmZpYBBIPI718hCihLDZ5KhKX
Supplier#000000710 f19YPvOyb
QoYwjKC,oPycpGfieBAcwKJo
Supplier#000000736
l6i2nMwVuovfKnuVgaSGK2rDy65DlAFLegiL7
Supplier#000000761
zlSLelQUj2XrvTTFnv7WAcYZGvvMTx882d4
Supplier#000000884 bmhEShejaS
Supplier#000000887 urEaTejH5POADP2ARrf
Supplier#000000935 ij98czM
2KzWe7dDTOxB8sq0UfCdvrX
Supplier#000000975 ,AC
e,tBpNwKb5xMUzeohxlRn, hdZJo73gFQF8y
Supplier#000001263 rQWr6nf8ZhB2TAiIDIvo5Io
Supplier#000001399 LmrocnIMSyYOWuANx7
Supplier#000001446
lch9HMNU1R7a0LIybsUodVknk6
Supplier#000001454 TOpimgu2TVXIjhiL93h,
Supplier#000001500 wDmF5xLxtQch9ctVu,
Supplier#000001602 uKNWIeafaM644
Supplier#000001626 UhxNRzUu1dtFmp0
Supplier#000001682 pXTkGxrTQVyH1Rr
Supplier#000001699 Q9C4rfJ26oijVPqqcqVXeRI
Supplier#000001700 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
hz2qWXWVjOyKhqPYMoEwz6zFkrTaDM
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,trBENywSArwg3DhB
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
Supplier#000005283
5fxYXxwXy,TQX,MqDC2hxzyQ
Supplier#000005300 gXG28YqpxU
Supplier#000005386 Ub6AAfHpWLWP
Supplier#000005426 9Dz2OVT1q
sb4BK71ljQ1XjPBYRPvO
Supplier#000005484 saFdOR
qW7AFY,3asPqiiAa11Mo22pCoN0BtPrKo
Supplier#000005505 d2sbjG43KwMPX
Supplier#000005506 On f5ypzoWgB
Supplier#000005516
XsN99Ks9wEvcohU6jRD2MeebQFf76mD8vovuY
Supplier#000005536
Nzo9tGkpgbHT,EZ4D,77MYKl4ah1C
Supplier#000005605 7Vj6Eil0mThqkM
Supplier#000005631
14TVrjlzo2SJEBYCDgpMwTlvwSqC
Supplier#000005730 5rkb0PSews
HvxkL8JaD41UpnSF2cg8H1
Supplier#000005736 2dq XTYhtYWSfp
Supplier#000005737 dmEWcS32C3kx,d,B95
OmYn48
Supplier#000005797
,o,OebwRbSDmVl9gN9fpWPCiqB UogvlSR
Supplier#000005836 tx3SjPD2ZuWGFBRH,
Supplier#000005875
lK,sYiGzB94hSyHy9xvSZFbVQNCZe2LXZuGbS
Supplier#000005974
REhR5jE,lLusQXvf54SwYySgsSSVFhu
Supplier#000005989 rjFY,5kgLpBu7c
Supplier#000006059 4m0cv8MwJ9yX2vlwI Z
Supplier#000006065 UiI2Cy3W4Tu5sLk
LuvXLRy6KihlGv
Supplier#000006070 TalC5m0pDrO6DZbngfmGmqe
Supplier#000006109
rY5gbfh3dKHnylcQUTPGCwnbe
Supplier#000006121
S92ycWwEzYYw4GspCBJN1WMuHhoZ
Supplier#000006215
j2iEbTsl,5PWdqWZ7k1yiISb7qtiiZljDIPEo
Supplier#000006217 RVN23SYT9jenUeaWGXUd
Supplier#000006274 S3yTZWqxTKUq g QQgcW9
AqhCkNZsW51hHuwU
Supplier#000006435
xIgE69XszYbnO4Eon7cHHO8y
Supplier#000006463 7 wkdj2EO49iotley2kmIM
ADpLSszGV3RNWj
Supplier#000006493 ojV
f,sNaB6Hm7r,fknDVTL63raJgAjZK
Supplier#000006521 b9 2zjHzxR
Supplier#000006607 3F 2e2gqD5u5B
Supplier#000006706
Ak4ga,ePu1QZ6C3qkrqjosaX0gxvqS9vkbe
Supplier#000006761
n4jhxGMqB5prD1HhpLvwrWStOLlla
Supplier#000006808 HGd2Xo
9nEcHJhZvXjXxWKIpApT
Supplier#000006858
fnlINT885vBBhsWwTGiZ0o22thwGY16h GHJj21
Supplier#000006872 XIDPiA7PLXCWK6SeEcld
Supplier#000006949 mLxYUJhsGcLtKe
,GFirNu183AvT
Supplier#000006985
PrUUiboQpy,OtgJ01Z4BxJQUyrw9c3I
Supplier#000007072 2tRyX9M1a
4Rcm57s779F1ANG9jlpK

Supplier#000007098
G3j8g0KC4OcbAu2OVoPHrXQWMCUdjq8wgCHOExu
Supplier#000007135 ls DoKV7V5ulfQy9V
Supplier#000007160
TqDGBULB3cTqIT6FKDvm9BS4e4v,zwYiQPb
Supplier#000007169 tEc95D2moN9S84nd55O,dlnW
Supplier#000007322 wr7dgte5q
MAjiY0uwmi3MyDkSMX1
Supplier#000007365 51xhROLvQMJ05DndtZWt
Supplier#000007398 V8eE6oZ00OFNU,
Supplier#000007402 4UVv58ery1rjmqSR5
Supplier#000007448 yhhpWiJi7EJ6Q5VCaQ
Supplier#000007477
9m9j0wfhWzCvVHxkU,PpAxwSH0h
Supplier#000007509
q8,V6LJRoHJjHcOuSG7aLTMg
Supplier#000007561 rMcFg2530VC
Supplier#000007789
rQ7cUcPrtudOyO3svNSkimqH6qrfWT2Sz
Supplier#000007801 69fi,U1r6enUb
Supplier#000007818 yhhc2CQec Jrvc8zqBi83
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,BfCXN5BIAb
Supplier#000008366
h778cEj14BuW9OEKlvPTWq4iwASR6EBBXN7zeS8
Supplier#000008423
RQhKnkAhR0DAr3Ix4Q1weMMn00hNe Kq
Supplier#000008480 4sSDA4ACReklNjEm5T6b
Supplier#000008532
Uc29q4,5xVdDOF87UZrxhr4xWS0ihEUXuh
Supplier#000008595 MH0iB73GQ3z UW3O
DbCbqmc
Supplier#000008610
SgVgP90vP452sUNTgzL9zKwXHXAzV6tV
Supplier#000008705
aE,trRNdPx,4yinTD9O3DebDIp
Supplier#000008742 HmPlQEzKCPEcTUL14,kKq
Supplier#000008841 I 85Lu1sekbg2xrSIzm0
Supplier#000008895
2cH4okfaLSZTTg8sKRbbJQxkmeFu2Esj
Supplier#000008967 2kwEHyMG
7FwozNImAUE6mH0hYtqYculJM
Supplier#000008972 w2vF6
D5YZO3visPXsqVfLADTK
Supplier#000009032 qK,trB6Sdy4Dz1BRUFNy
Supplier#000009147 rOAuryHxpZ9eOvx
Supplier#000009252 F7cZaPUHwh1
ZKyj3xmAVWC1XdP ue1p5m,i
Supplier#000009278 RqYTzgxj93CLX
0mcYfCENOefD
Supplier#000009327 uoqMdf7e7Gj9dbQ53
Supplier#000009430 igRqmneFt
Supplier#000009567
r4Wfx4c3xsEAjcGj71HHZByornl D9vrztXlv4
Supplier#000009601
51m637bO,Rw5DnHWFUvLacRx9
Supplier#000009709
rRnCbHYgDgl9PZYnyWKVYSUW0vKg
Supplier#000009753 wLhVEcRmd7PkJF4FBnGK7Z
Supplier#000009796 z,y4Idmr15DOvPUqYG
Supplier#000009799 4wNjXGa4OKWl
Supplier#000009811 E3iuyq7UnZxU7oPZIe2Gu6
Supplier#000009812
APFRMy3lCbgFga53n5t9DxzFPQPgnjrGt32
Supplier#000009862 rJzweWeN58
Supplier#000009868 ROjGgx5gvtkmnUUoeyy7v
Supplier#000009869
ucLqxzrpBTRMewGSM29t0rNTM30g1Tu3Xgg3mKag
Supplier#000009899 7XdpAHrzr1t,UQFZE
Supplier#000009974
7wJ,J5DKcxSU4Kp1cQLpbcAvB5AsvKT

204 rows processed.
Query Processed in 1.09 seconds.

-- Q21

```
select * from (
select
s_name,
count(*)  numwait
from
supplier,
lineitem l1,
orders,
nation
where
s_suppkey = l1.l_suppkey
and o_orderkey = l1.l_orderkey
and o_orderstatus = 'F'
and l1.l_receiptdate > l1.l_commitdate
and exists (
select
*
```

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

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

-- Q22

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

CNTRYCODE	NUMCUST	TOTACCTBAL
13	888.00	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 1.39 seconds.

Appendix E Seed and Input Parameters

E.1 Seed

506082536

E.2 qp3.0

```
14	1997-01-01
2	34	TIN	EUROPE
9	chiffon
20	pink	1997-01-01	ARGENTINA
6	1996-01-01	0.06	25
17	Brand#45	SM JAR
18	315
8	ROMANIA	EUROPE	STANDARD BRUSHED STEEL
21	UNITED KINGDOM
13	pending	deposits
3	MACHINERY	1995-03-19
22	11	13	10	14	30	29	17
16	Brand#24	PROMO BURNISHED	38	35	16	1	42	5	8	19
4	1996-01-01
11	SAUDI ARABIA	0.0000000333
15	1993-01-01
1	84
10	1994-12-01
19	Brand#14	Brand#51	Brand#55	1	11	28
5	AMERICA	1996-01-01
7	JAPAN	ROMANIA
12	MAIL	RAIL	1997-01-01
```

E.3 qp3.1

```
21	MOROCCO
3	BUILDING	1995-03-05
18	313
5	ASIA	1996-01-01
11	INDIA	0.0000000333
7	EGYPT	IRAQ
6	1996-01-01	0.03	24
20	brown	1996-01-01	MOZAMBIQUE
17	Brand#42	SM PKG
12	TRUCK	RAIL	1997-01-01
16	Brand#54	SMALL PLATED	24	6	27	42	50	34	18	12
15	1996-01-01
13	pending	deposits
10	1993-09-01
2	22	STEEL	AFRICA
8	IRAQ	MIDDLE EAST	STANDARD PLATED COPPER
14	1997-01-01
19	Brand#11	Brand#44	Brand#44	6	12	24
9	blue
22	30	17	10	24	15	23	32
1	92
4	1993-10-01
```

E.4 qp3.2

```
6	1996-01-01	0.08	24
17	Brand#44	LG CASE
14	1997-01-01
16	Brand#44	LARGE BRUSHED	4	27	25	18	9	1	14	46
```

19 Brand#23 Brand#22 Brand#43 2 13 20
10 1994-06-01
9 aquamarine
2 10 BRASS EUROPE
15 1994-01-01
8 CANADA AMERICA STANDARD ANODIZED COPPER
5 EUROPE 1996-01-01
22 15 21 33 20 18 31 14
12 RAIL TRUCK 1997-01-01
7 VIETNAM CANADA
13 pending deposits
18 314
1 100
4 1996-05-01
20 medium 1994-01-01 ETHIOPIA
3 MACHINERY 1995-03-21
11 VIETNAM 0.0000000333
21 GERMANY

E.5 qp3.3

8 ROMANIA EUROPE PROMO POLISHED COPPER
5 MIDDLE EAST 1997-01-01
4 1994-01-01
6 1997-01-01 0.06 25
17 Brand#41 LG JAR
7 JORDAN ROMANIA
1 108
18 312
22 30 24 19 12 33 28 21
14 1993-01-01
9 violet
10 1993-04-01
15 1996-01-01
11 INDONESIA 0.0000000333
20 turquoise 1997-01-01 SAUDI ARABIA
2 47 NICKEL AMERICA
21 UNITED STATES
19 Brand#25 Brand#55 Brand#42 7 14 28
13 pending packages
16 Brand#24 STANDARD ANODIZED 14 48 50 1 32 34 11 13
12 AIR TRUCK 1993-01-01
3 BUILDING 1995-03-07

E.6 qp3.4

5 AFRICA 1997-01-01
21 MOZAMBIQUE
14 1993-01-01
19 Brand#22 Brand#43 Brand#32 2 15 24
15 1994-01-01
17 Brand#43 LG PKG
12 REG AIR TRUCK 1993-01-01
6 1997-01-01 0.03 24
4 1996-08-01
9 spring
8 IRAQ MIDDLE EAST PROMO BURNISHED COPPER
16 Brand#14 MEDIUM PLATED 35 19 34 43 32 12 37 10
11 RUSSIA 0.0000000333
2 35 TIN EUROPE
10 1994-01-01
18 313
1 116
13 unusual packages
7 ETHIOPIA IRAQ
22 25 17 18 16 12 19 33
3 HOUSEHOLD 1995-03-23
20 green 1996-01-01 IRAN

E.7 qp3.5

21 INDONESIA
15 1997-01-01
4 1994-05-01
6 1997-01-01 0.09 24
7 RUSSIA CANADA
16 Brand#44 ECONOMY POLISHED 12 40 9 21 37 24 28 30
19 Brand#34 Brand#21 Brand#31 7 16 20
18 315
14 1993-01-01
22 18 15 22 16 24 11 31
11 IRAN 0.0000000333
13 unusual packages
3 AUTOMOBILE 1995-03-09
1 63
2 23 COPPER AMERICA
5 AMERICA 1997-01-01
8 CANADA AMERICA ECONOMY BRUSHED COPPER
20 royal 1994-01-01 ALGERIA
12 FOB TRUCK 1993-01-01
17 Brand#44 MED CASE
10 1994-10-01
9 seashell

E.8 qp3.6

10 1993-07-01
3 HOUSEHOLD 1995-03-25
15 1994-01-01
13 unusual packages
6 1997-01-01 0.06 25
8 SAUDI ARABIA MIDDLE EAST ECONOMY POLISHED TIN
9 rose
7 KENYA SAUDI ARABIA
4 1996-12-01
11 UNITED KINGDOM 0.0000000333
22 24 20 32 15 17 14 22
18 312
12 MAIL AIR 1994-01-01
1 71
5 EUROPE 1997-01-01
16 Brand#24 SMALL ANODIZED 24 33 3 43 9 46 2 35
2 11 BRASS MIDDLE EAST
14 1993-01-01
19 Brand#32 Brand#13 Brand#35 3 17 27
20 cornsilk 1993-01-01 KENYA
17 Brand#41 MED JAR
21 ARGENTINA

E.9 qp3.7

18 314
8 JAPAN ASIA ECONOMY BURNISHED TIN
20 olive 1996-01-01 EGYPT
21 CHINA
2 49 NICKEL AMERICA
4 1994-09-01
22 22 18 28 32 19 24 26
17 Brand#43 MED PKG
1 79
11 IRAQ 0.0000000333

9 pink
19 Brand#34 Brand#41 Brand#24 8 18 23
3 AUTOMOBILE 1995-03-11
13 unusual packages
5 MIDDLE EAST 1993-01-01
7 FRANCE JAPAN
10 1994-04-01
16 Brand#14 LARGE BURNISHED 16 48 34 10 41 38 26 40
6 1993-01-01 0.04 25
14 1994-01-01
15 1997-01-01
12 TRUCK MAIL 1994-01-01

E.10 qp3.8

19 Brand#41 Brand#34 Brand#23 3 19 20
1 87
15 1995-01-01
17 Brand#45 JUMBO CASE
5 AFRICA 1993-01-01
8 EGYPT MIDDLE EAST LARGE BRUSHED TIN
9 orange
12 RAIL MAIL 1994-01-01
14 1994-01-01
7 UNITED KINGDOM EGYPT
4 1997-04-01
3 FURNITURE 1995-03-27
20 azure 1995-01-01 ROMANIA
16 Brand#44 PROMO POLISHED 27 19 22 21 15 40 25 32
6 1993-01-01 0.09 24
22 34 13 24 16 23 29 26
10 1993-02-01
13 unusual requests
2 36 TIN MIDDLE EAST
21 IRAN
18 315
11 UNITED STATES 0.0000000333

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;

SELECT COUNT(*) FROM CUSTOMER;

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

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

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

/* OCI handles */

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

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

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

void usage() {

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



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

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

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

```
                                        2048,OCI_HTYPE_ENV);
   fprintf(stderr,"%s\n",msg);
   break;
  }

  /* Rollback just in case */

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

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

  SQLexit();

  exit(1);
}

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

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

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

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

  strcpy(logname, argv[1]);

  /* Process  optional parameters */

  argc -= 1;
  argv += 1;

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

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

  SQLinit();

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

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

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

    switch (retcode) {

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

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

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

      /* Executes the query */
      SQLexec();

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

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

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

 /* Get Timing for the last query */

 tr_end = gettime();

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

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

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

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

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

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

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

 fflush(rep);
 fflush(logfile);

 /* Close the query template file */

 fclose(qtemp);

 /*  Disconnect from ORACLE. */

 SQLexit();
 exit(0);
}


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

void SQLinit() {
 int i;

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

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

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

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

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

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

 /* get username and password */

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

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


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

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

OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI_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 */


   strcpy(qnp,qn);

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

 s_tr_start = gettime();

 /* prepare the statement */

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

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

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

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


OCIaget(curq,OCI_HTYPE_STMT,&stmttyp,NULL,OCI_ATTR_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 %s processed.\n", rows_ret,
            rows_ret == 1 ? "row" : "rows");

}


int get_statement()
{

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

  /* Reset statement buffer */

  stmt[0] = '\0';

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

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

   /* remove blanks */

   str = line;

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

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

   strcat(stmt, str);

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

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

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


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

void *memalloc(size)
     int size;
{

  void *tmp;

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



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

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

  fprintf(logfile, "\n");

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

    /* extract the column name */

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

    /* format the output a little */
```

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

   /* do a little bit of formatting */

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

  fprintf(logfile,"\n");
}


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

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

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

    len = 0;

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

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

      /* do a little bit of formatting */

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

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


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

void remove_newline(str)
     char *str;
{

  char *p;

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

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 dbload.sh

```
#!/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
LD3DAPOP=${OUT_DIR}/populate.out

echo Start TPC-H load SEQUENCE NUMBER: $RUN_ID >
$SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE
/usr/bin/rsh $HOSTNAME "move
C:\oracle\product\10.2.0\admin\tpch\bdump\alert_tpch.log
e:\audit\alert_tpch_bef.log" >> $SCRIPT_LOG_FILE
STIME=`$GTIME`
echo "Start: timed load portion `date`" >> $SCRIPT_LOG_FILE
. ./dapop_10gR2.sh >> $LD3DAPOP
#/usr/bin/rsh $HOSTNAME "shutdown -r -t 0" >>
$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 "End: timed load portion `date`" >>
$SCRIPT_LOG_FILE
#
# restart SUT and start DB, then generate seed
#
echo "End TPC-H load SEQUENCE NUMBER:
$RUN_ID `date`" >> $SCRIPT_LOG_FILE
```

F.9 Genseed.sh

```
#!/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`

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

```
SCRIPT_LOG_FILE=${OUT_DIR}/gen_seed.out
$KIT_DIR/audit/gen_seed.sh $KIT_DIR/audit/seed

echo Generated seed: `cat $KIT_DIR/audit/seed` >>
$SCRIPT_LOG_FILE
echo "End: timed load portion `date`" >> $SCRIPT_LOG_FILE
```

F.10 runTPCH1

```
#!/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`

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

SCRIPT_LOG_FILE=${OUT_DIR}/main_run1.out
RDB_TABLES=${OUT_DIR}/rdbtablest
FIRST_TEN=${OUT_DIR}/firstten
RITEST=${OUT_DIR}/ritest
echo Start TPC-H Benchmark run 1 SEQUENCE NUMBER: $RUN_ID
> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE
/usr/bin/rsh $HOSTNAME "move
C:\oracle\product\10.2.0\admin\tpch\bdump\alert_tpch.log
e:\audit\alert_tpch_bef_run1.log" >> $SCRIPT_LOG_FILE
STIME=`$GTIME`

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

runTPCHpt ${SCALE_FACTOR} 1 ${RUN_ID}

/usr/bin/rsh $HOSTNAME "copy
C:\oracle\product\10.2.0\admin\tpch\bdump\alert_tpch.log
e:\audit\alert_tpch_aft_run1.log" >> $SCRIPT_LOG_FILE
echo "End TPC-H Benchmark run 1 SEQUENCE NUMBER: $RUN_ID
`date`" >> $SCRIPT_LOG_FILE

#!/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`

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

SCRIPT_LOG_FILE=${OUT_DIR}/main_run2.out

echo Start TPC-H Benchmark run2 SEQUENCE NUMBER: $RUN_ID
> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE
/usr/bin/rsh $HOSTNAME "move
C:\oracle\product\10.2.0\admin\tpch\bdump\alert_tpch.log
e:\audit\alert_tpch_bef_run2.log" >> $SCRIPT_LOG_FILE
STIME=`$GTIME`

runTPCHpt ${SCALE_FACTOR} 3 ${RUN_ID}

/usr/bin/rsh $HOSTNAME "copy
C:\oracle\product\10.2.0\admin\tpch\bdump\alert_tpch.log
e:\audit\alert_tpch_aft_run2.log" >> $SCRIPT_LOG_FILE
echo "End TPC-H Benchmark run2 SEQUENCE NUMBER: $RUN_ID
`date`" >> $SCRIPT_LOG_FILE
```

F.11 runTPCHpt

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

DSS_QUERY=${KIT_DIR}/queries
export DSS_QUERY

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

TPCD_BIN=${KIT_DIR}/audit/bin

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

DF=/dev/null
HID=1
INTERVAL=60
```

```
COUNT=1200

# The defaults

QPROG=${QEXEC}/qexec

usage () {

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

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


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

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

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

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

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

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

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

 ${QGEN} -c -r `cat $SEED_FILE` -p 0 -s ${SF} -l
$QUERY_PARAMETER > ${QRY_FILE}
START=`$GTIME`
echo "Start Power Test - RUN:${PARA} SEQUENCE:${RUN_ID}
Execution Starts $START, `date`" >>  $SCRIPT_LOG_FILE
echo "" >> $SCRIPT_LOG_FILE

# Execute UF1

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

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

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

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

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

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

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

# Execute UF2

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

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

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

UF2_TIME=`echo $UF2_END - $UF2_START | bc`
```

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


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

echo "End TPC-H Power Test - RUN:${PARA}
SEQUENCE:${RUN_ID}, $END, $EDATE" >> $SCRIPT_LOG_FILE
MEA_INT=`echo $END - $START | bc`
echo "Elapsed Time for TPC-H Power Test - RUN:${PARA}
SEQUENCE:${RUN_ID} is $MEA_INT" >> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE
${KIT_DIR}/audit/abridge.pl ${TPCD_LOG_FILE}
i=$START_SET
PSEED=`cat $SEED_FILE`

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

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

  i=`expr $i + 1`
done
TH_START_D=`date`
TH_START_T=`${GTIME}`
echo >> $SCRIPT_LOG_FILE

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

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

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

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

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

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

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

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

F.12 runTPCHus

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

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

DSS_QUERY=${KIT_DIR}/queries
export DSS_QUERY

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

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


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

GTIME=${SRC_DIR}/gtime
HID=1

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

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

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

  # Execute UF1

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

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

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

  # Execute UF2

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

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

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

print > /tmp/th_pipe2
```

F.13 runuf1.sh

```
#!/bin/ksh
#
# $Header: runuf1.sh 25-oct-2001.15:56:04 mpoess Exp $
#
# runuf1.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation.  All rights reserved.
#
#    NAME
#      runuf1.sh - <one-line expansion of the name>
#
# DESCRIPTION
#    runuf1.sh -l [<path name for reports>] -u [<uid/passwd>]
#             -p [<program>] <run_id> <scale factor> <pair number>
#             <parallelism>
# USAGE
#    To execute UF1.
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      10/25/01 - change default directory for update sets
#    mpoess      10/17/01 - add support for external tables
#    mpoess      08/15/99 - Creation
#    mpoess      08/15/99 - Creation
#
#
. $KIT_DIR/env
O=${ORACLE_HOME}
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
UTILS_DIR=${KIT_DIR}/utils
LOG_DIR=${UPDATE_DIR}/log
GTIME=${UTILS_DIR}/gtime
SF=${SCALE_FACTOR}
PAR_HINT=64
#was 32

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 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 ff1
access parameters
(
            records delimited by newline
            nobadfile
            nologfile
```

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

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.14 runuf2.sh

```
#!/bin/ksh
#
# $Header: runuf2.sh 25-oct-2001.15:56:05 mpoess Exp $
#
# runuf2.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation.  All rights reserved.
#
#   NAME
#     runuf2.sh - <one-line expansion of the name>
#
# DESCRIPTION
#    runuf2.sh [-u <uid/passwd to login>] [-p <program>] <run_id>
#              <scale factor> <pair number> <parallelism>
# USAGE
#    To execute UF2.
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      10/25/01 - change default directory for update sets
#    mpoess      10/17/01 - add support for external tables
#    mpoess      08/15/99 - Creation
#    mpoess      08/15/99 - Creation
#
. $KIT_DIR/env
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
UTILS_DIR=${KIT_DIR}/utils
GTIME=${UTILS_DIR}/gtime
LOG_DIR=${UPDATE_DIR}/log
PAR_HINT=64
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 table temp_okey_et;
drop table temp_okey;

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

alter table temp_okey_et parallel ${PAR_HINT};

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

create unique index i_temp_okey on temp_okey (t_orderkey) nologging
compute statistics;

analyze table temp_okey estimate statistics sample 2 percent;


alter session  force parallel dml parallel ${PAR_HINT};
alter session set isolation_level=serializable;
alter session set optimizer_index_cost_adj = 1;
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`"
```

Appendix G Price Quotes

From: MaryBeth Pierantoni [mary.beth.pierantoni@oracle.com]

Sent: Tuesday, April 24, 2007 1:44 PM

To: Broodbakker, Mario

Subject: Oracle Pricing

Product	Price	Quantity	Extended Price
Oracle Database 10g Enterprise Edition, Named User Plus, 3 years	**$10,000**	**16**	**$160,000**
Partitioning, Named User Plus, 3 years	**$2,500**	**16**	**$40,000**
Oracle Database Server Support Package for 3 years	**$2,000**	**3**	**$6,000**
Oracle Mandatory E-Business Discount			**<$30,900>**
Oracle TOTAL			**$175,100**

*** 16 = 0.50 * 32. Explanation: For the purposes of counting the number of processors which require licensing, an Intel multicore chip with "n" cores shall be determined by multiplying "n" cores by a factor of 0.50.**

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