TPC Benchmark™ H

Full Disclosure Report for

FUJITSU

PRIMEPOWER 2500

Using Oracle Database 10g Enterprise Edition

Jan. 14, 2004

Second Edition

Second Edition Jan.14, 2004

Fujitsu Limited believes that the information in this document is accurate as of the publication date. The information in this document is subject to change without notice. We assume no responsibility for any errors that may appear in this document. The pricing information in this document is believed to accurately reflect the current prices as of the publication date. However, we provide no warranty of the pricing information in this document.

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

All performance data contained in this report were obtained in a rigorously controlled environment. Results obtained in other operating environments may vary significantly. We do not warrant or represent that a user can or will achieve similar performance. No warranty of system performance or price/performance is expressed or implied in this report.

Preface

The Transaction Processing Performance Council (TPC), of which Fujitsu Limited is a member, is an organization of computer companies, dedicated to the development of objective, industry-wide performance metrics in the area of transaction processing. Fujitsu Limited is involved in this effort, participating on the council and utilizing TPC benchmarks in performance evaluation.

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. 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:

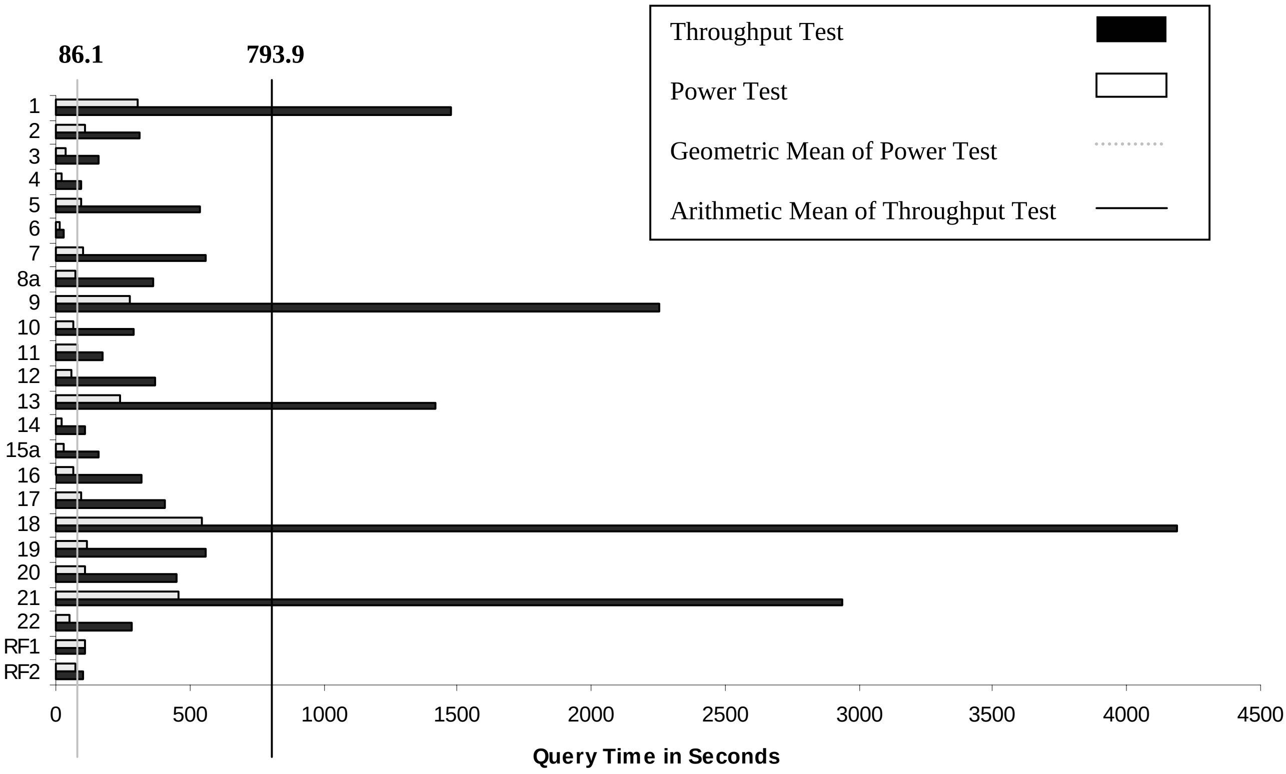
- Give answers to real-world business questions
- Simulate generated ad-hoc queries
- Are far more complex than most OLTP transactions
- Include a rich breadth of operators and selectivity constraints
- Generate intensive activity on the part of the database server component of the system under test
- 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 performance metric reported by TPC-H is called the TPC-H Composite Query-per-Hour Performance Metric (QphH@Size), and reflects multiple aspects of the capability of the system to process queries. These aspects include the selected database size against which the queries are executed, the query processing power when queries are submitted by a single stream, and the query throughput when queries are submitted by multiple concurrent users. The TPC-H Price/Performance metric is expressed as $/QphH@Size.

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.

FUJITSU ORACLE	**PRIMEPOWER 2500** **With Oracle Database 10g**	**TPC-H REV 2.0** **EXECUTIVE SUMMARY** **Report Date: Jan. 14, 2004**
Total System Cost	Composite Query per Hour Metric	Price/Performance
$ 4,861,985	**34,492.5** QphH@1000GB	**$141** $/QphH@1000GB

Database Size	Database Manager	Operating System	Other Software	Availability Date
1000GB	**Oracle Database 10g Enterprise Edition**	**Solaris 9**	**Veritas Volume Mgr.3.5.0**	**March 8, 2004**



Database Load time = 02:18:34	Load Includes Backup: N	Total Data Storage/database Size=43.68
RAID (base tables): N	RAID (Base Tables & auxil. Data structures): N	RAID (A11): Y

System Configuration:	PRIMEPOWER 2500
Processors:	64 Fujitsu SPARC64 V 1.3 GHz with 2 MB L2 cache
Memory:	256 GB
Disks:	5 internal disks 36 GB, 64 FibreCat S80 arrays with 640 disks 73 GB
Total Storage:	43,679.04 GB (1GB defined as 2^30 bytes)

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

FUJITSU

ORACLE

PRIMEPOWER 2500

With Oracle Database 10g

TPC-H REV 2.0
EXECUTIVE SUMMARY

Report Date: Jan. 14, 2004

Description	Part Number	Source	Unit Price	Qty	Ext. Price	3 Yr. Maint.
Server Hardware						
PRIMEPOWER2500(1.3GHz/2MB$)(For 8-64CPU)- Solaris9	PW2K0BP7U	1	504,450	1	504,450	457,776
Systemboard	PW2K7SB11U	1	37,500	8	300,000	
1.3GHz/2MB$ CPU Module Expansion (2x)	PW2K1B12U	1	42,510	32	1,360,320	
8GB Additional Memory (1.3GHz/2MB$)	PW2K2M41U	1	14,800	32	473,600	
36GB Hard Disk Drive, AL8LE	PW0R3D24	1	1,000	5	5,000	
PCI/DISK BOX	PW2K7BD1U	1	27,500	16	440,000	
I/O Rack	PW2K7RK3U	1	24,500	2	49,000	
I/O Rack Joint Kit	PW2K7RK91U	1	9,500	1	9,500	
I/O Rack Joint Kit	PW2K7RK92U	1	9,500	1	9,500	
Fibre Channel Card(2Gb)	PW028FC3	1	2,795	64	178,880	
System Expansion Kit (w/ Solaris 9 RTU)	PW2K7ER6U	1	4900	1	4900	
Fibre Channel cable	DCL-FCA20U	1	780	64	49920	
PRIMEPOWER 100 for System Console	PW087MC4AU	1	13,500	1	13,500	
				Subtotal	**3,398,570**	**457,776**
Server Software						
Solaris 9 CD set PRIMEPOWER		1	100	1	100	
Sun ONE Studio 7, Compiler Collection Slim Kits 1 User Slim Kit	D23QBB0H1H	1	987	1	987	
Volume Manager, Solaris, v3.5, License, Tier 4C	A08974F-M0000	2	39,126.75	1	39,126.75	
Volume Manager,Solaris,v3.5 Tier 4C Extended Support, 3 Yr 24x7	W08974F-M00236	2		3		35,304.75
Storage Solution, Solaris v3.5 Maintenance Pack 1, English, Media Kit	N09665F	2	65	1	65	
Volume Manager, Solaris, Administrator's Guide, v3.5 English Manual	N08836F	2	32.50	1	32.50	
Oracle Database 10g Enterprise Edition for 3 years, Named User Plus		3	10,000	64	640,000	
Partitioning for 3 years, Named User Plus		3	2,500	64	160,000	
Database Server Support Package for 3 years		3	6,000	1		6,000
Oracle Mandatory E-Business Discount (license and support)		3			-161,200	
				Subtotal	**679,111.25**	**41,304.75**
Storage						
FC-S80 Basis Shelf	D:S80-Base	4	3,650	64	233,600	225,000
S80 RAID Controller 2 Gbit/s	D:S80FC-RDM	4	9,675	64	619,200	80,000
S80 Disk, 73GB 10.000rpm	D:S80-HD7310	4	1,700	640	1,088,000	
PRIMECENTER Rack 38 HE	D:GPRAC-BG52	4	2,350	6	14,100	
				Subtotal	**1,954,900**	**305,000**
Large Configuration Discount and Support Prepayment*					-1,899,602	-75,075
Notes:		**Total**			**$4,132,979**	**$729,006**
Source: 1=Fujitsu		**3 Yr. Cost**			**$4,861,985**	
2=Veritas, contact: Michael Mawson (see Appendix G)		**QphH@1000GB**			**34,492.5**	
3=Oracle, contact: MaryBeth Pierantoni (see Appendix G)		**$/QphH@1000GB**			**$141**	
4=Fujitsu Siemens, contact: Jürgen Binder (see Appendix G)						
*All discounts are based on US list prices and for similar quantities and configurations						

Audited by: Francois Raab, InfoSizing, Inc. (www.sizing.com)

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

Numerical Quantities

Measurement Results:

Database Scale Factor	1000 GB	
Total Data Storage / Database Size	43.68	
Start of Database Load Time	05.09.2003	08:08:28
End of Database Load Time	05.09.2003	10:27:02
Database Load Time	2:18:34	
Query Streams for Throughput Test	7	
TPC-H Power	41,829.5	
TPC-H Throughput	28,442.4	
TPC-H Comp. Query-per-Hour-Rating (QphH@1000GB)	34,492.5	
Total System Prize over 3 Years (US$)	4,861,985	
TPC-H Price/Performance Metric (US$/QphH@1000GB)	141	

Measurements Intervals:

Measurement Interval in Throughput Test	19,492	Seconds

Duration of Stream Execution:

Stream ID	Seed	Start Date	Start Time	End Date	End Time	Duration
Stream 00	905102702	05. Sep	16:54:50	05. Sep	17:47:18	00:52:28
Stream 01	905102703	05. Sep	17:47:39	05. Sep	22:36:06	04:48:27
Stream 02	905102704	05. Sep	17:47:39	05. Sep	22:34:56	04:47:17
Stream 03	905102705	05. Sep	17:47:39	05. Sep	22:36:15	04:48:36
Stream 04	905102706	05. Sep	17:47:39	05. Sep	22:29:30	04:41:51
Stream 05	905102707	05. Sep	17:47:39	05. Sep	22:45:16	04:57:37
Stream 06	905102708	05. Sep	17:47:39	05. Sep	22:41:27	04:53:48
Stream 07	905102709	05. Sep	17:47:39	05. Sep	22:47:51	05:00:12
Refresh		05. Sep	22:47:51	05. Sep	23:12:31	00:24:40

TPC-H Timing Intervals (in seconds)

	1	2	3	4	5	6	7	8a	9	10	11	12
Stream 00	306.0	105.6	33.3	23.3	94.7	13.8	101.4	71.9	276.1	64.3	81.3	61.2
Stream 01	1161.7	315.4	120.0	38.6	645.9	19.5	660.4	501.4	2346.6	369.6	172.4	304.7
Stream 02	1424.6	312.7	127.0	114.9	606.9	77.1	530.6	387.3	2110.3	289.3	144.3	370.9
Stream 03	1351.7	267.5	104.0	112.3	662.1	24.1	580.2	432.4	1931.0	386.7	203.1	458.4
Stream 04	1417.4	299.9	180.1	86.1	302.3	18.4	685.4	279.2	1960.1	387.5	176.2	358.5
Stream 05	1671.0	375.3	160.7	98.3	709.5	21.5	643.2	364.3	2084.1	392.9	160.4	485.2
Stream 06	1661.7	351.8	260.1	93.9	696.0	16.5	669.3	235.6	2351.3	165.1	147.2	532.1
Stream 07	1633.0	283.1	165.1	94.9	143.0	17.1	125.8	341.1	3009.8	67.7	211.5	99.2
Minimum	1161.7	267.5	104.0	38.6	143.0	16.5	125.8	235.6	1931.0	67.7	144.3	99.2
Average	1474.4	315.1	159.6	91.3	538.0	27.7	556.4	363.0	2256.2	294.1	173.6	372.7
Maximum	1671.0	375.3	260.1	114.9	709.5	77.1	685.4	501.4	3009.8	392.9	211.5	532.1

	13	14	15a	16	17	18	19	20	21	22	RF1	RF2
Stream 00	240.3	23.0	31.2	63.8	95.0	548.3	113.5	109.6	459.1	49.9	107.7	73.1
Stream 01	1538.1	125.9	166.7	327.1	299.9	3855.2	496.3	525.8	2989.3	326.1	113.2	74.2
Stream 02	1365.3	120.1	172.5	398.2	719.7	3877.1	618.4	301.8	2919.0	249.0	108.2	184.9
Stream 03	1452.5	112.0	217.2	324.5	450.2	4020.4	609.2	289.6	2987.8	339.5	110.9	91.0
Stream 04	1468.8	106.4	172.5	345.5	275.8	3923.4	536.1	830.5	2785.2	315.6	112.4	82.8
Stream 05	1481.4	130.7	179.6	338.2	367.2	3726.2	504.2	245.7	3463.5	253.5	110.2	85.0
Stream 06	1394.8	124.6	196.2	432.7	390.0	4047.7	585.5	281.5	2759.9	233.8	116.6	85.9
Stream 07	1203.8	28.0	33.4	71.5	366.3	5879.4	589.4	689.3	2673.7	284.6	107.3	96.7
Minimum	1203.8	28.0	33.4	71.5	275.8	3726.2	496.3	245.7	2673.7	233.8	107.3	74.2
Average	1415.0	106.8	162.6	319.7	409.9	4189.9	562.7	452.0	2939.7	286.0	111.2	100.1
Maximum	1538.1	130.7	217.2	432.7	719.7	5879.4	618.4	830.5	3463.5	339.5	116.6	184.9

INFOSIZING

TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL CERTIFIED AUDITOR

Test Sponsors: Ray Glasstone
Manger, DSS Performance.
Oracle Corporation
100 Oracle Parkway
Redwood Shores, CA 94065

Shin’ichi Kurogi
Manager, Server Business Devpt. Ctr.
Fujitsu Limited
Daiichi-Seimei Bldg. 2-7-1
Nishi-Shinjuku, Tokyo, Japan

November 7, 2003

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

Platform:	**PRIMEPOWER 2500**
Database Manager:	**Oracle Database 10g Enterprise Edition**
Operating System:	**Solaris 9**

The results were:

CPU (Speed)	Memory	Disks	**QphH@1000GB**
	PRIMEPOWER 2500		
64 x SPARC64 V (1.3 GHz)	2 MB L2-Cache/cpu 256 GB Main	640 x 73 GB 5 x 36 GB	**34,492.5**

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

- The database records were defined with the proper layout and size
- The database population was generated using DBGEN
- The database was properly scaled to 1 TB and populated accordingly
- The compliance of the database auxiliary data structures was verified
- The database load time was correctly measured and reported
- The required ACID properties were verified and met

- The query input variables were generated by QGEN
- The query text was produced using minor modifications and the approved variants 8a and 15a
- The execution of the queries against the SF1 database produced compliant answers
- A compliant implementation specific layer was used to drive the tests
- The throughput tests involved 7 query streams
- The ratio between the longest and the shortest query was such that no query timing was adjusted
- The execution times for queries and refresh functions were correctly measured and reported
- The repeatability of the measured results was verified. A failure during the second run of the benchmark required the execution of a third run, from which the reported results were collected.
- At least 8 hours of database log was configured
- The system pricing was verified for major components and maintenance
- The major pages from the FDR were verified for accuracy

Additional Audit Notes:

None.

Respectfully Yours,

François Raab
President

Contents

1 General Items

1.1 Benchmark Sponsor

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

This TPC-H benchmark is sponsored by Fujitsu Limited and Oracle Corp.

1.2 Parameter Settings

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

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

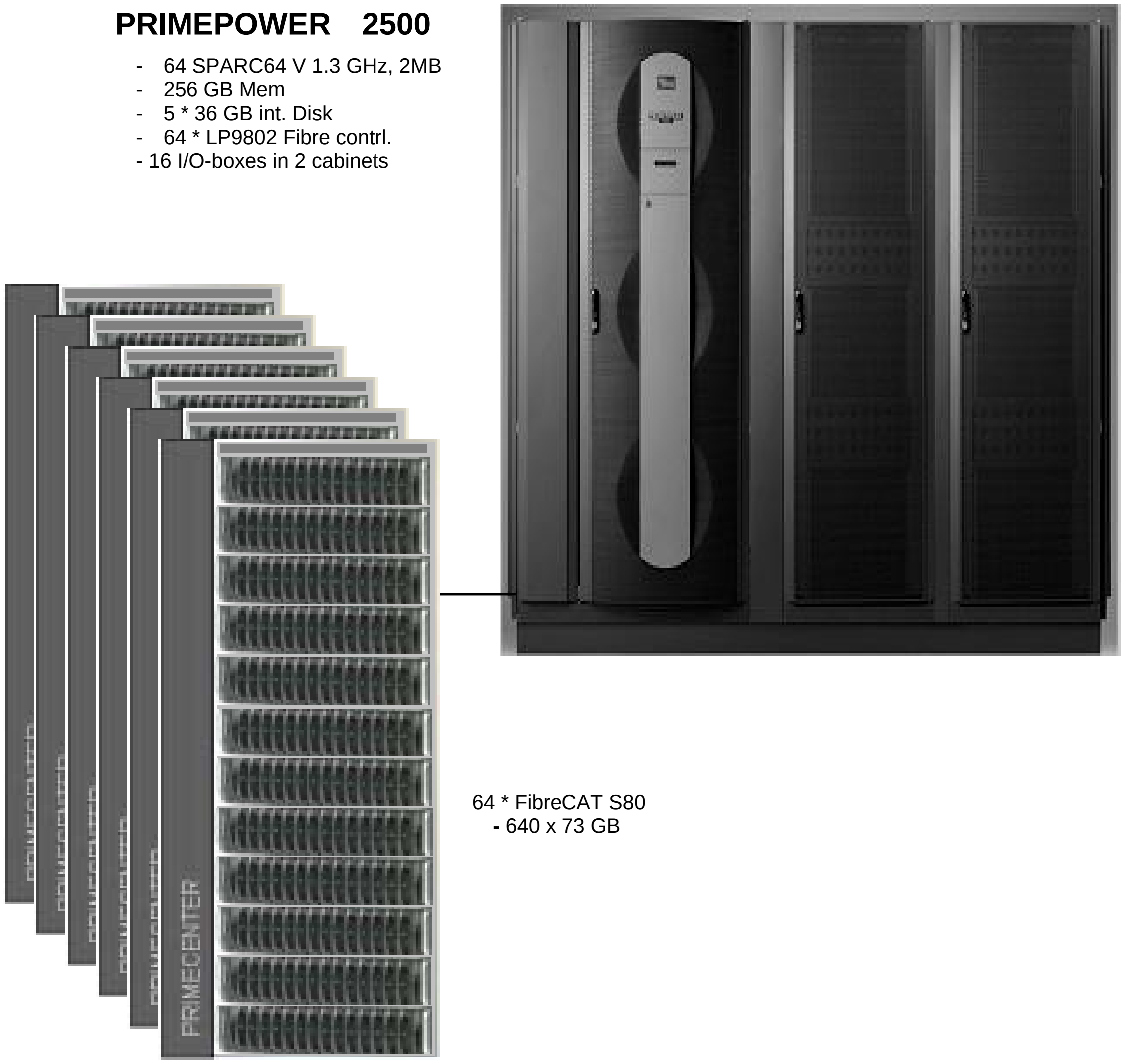
Appendix A contains the Solaris and Oracle parameters used in this benchmark.

1.3 Configuration Diagram

Provide diagrams of both the measured and priced configurations, accompanied by a description of the differences.

PRIMEPOWER 2500, configured with:

- 64 SPARC64 V 1.3 GHz processors
- 256 GB memory
- 5 * 36 GB internal disks
- 640 * 73 GB disks
- 16 I/O boxes in 2 cabinets
- 64 * FibreCAT S80 Storage array
- 64 * LP9802 Fibre Channel controllers



The previous description is for the priced configuration. There were additional 5 internal disks of 36 GB each and 64 S80 disks of 73 GB each in the measured configuration that were unused in this benchmark. Sufficient proof that the additional disks were unused was given to the auditor. The diagram is the same for both the priced and measured configurations.

2 Clause 1 Logical Database Design

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

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

2.1 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. Column ordering was changed for some tables. Refer to the table create statements in Appendix B for further details.

2.2 Horizontal Partitioning

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

Horizontal partitioning was used for all tables except NATION and REGION. Refer to the table/index create statements in Appendix B for more details.

2.3 Replication

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

No replication was used.

3 Clause 2 Queries and Refresh Functions

3.1 Query Language

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

SQL was the query language used to implement all queries.

3.2 Verifying Method for Random Number Generation

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

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

3.3 Generating Values for Substitution Parameters

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

The supplied QGEN version 2.1.8 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 definitions or approved variants in order to obtain executable query text, these modifications must be disclosed and justified. The justification for a particular minor query modification can apply collectively to all queries for which it has been used. The output data for the power and throughput tests must be made available electronically upon request.

Appendix C contains the qualification 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 D contains the seed and query substitution parameters.

3.6 Query Isolation Level

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

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

3.7 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 driver code included in Appendix E.

4 Clause 3 Database System Properties

4.1 ACID Properties

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

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

4.2 Atomicity

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

4.2.1 Completed Transaction

Perform the ACID Transaction 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.

4.2.2 Aborted Transaction

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

1. The total price from the 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.

4.3.1 Consistency Test

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

1. The consistency of the ORDERS and LINEITEM tables was verified based on a sample of order keys.
2. 100 ACID Transactions were submitted by each of nine 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 the proper order.

4.4.1 Read-Write Conflict with Commit

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

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

4.4.2 Read-Write Conflict with Rollback

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

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

4.4.3 Write-Write Conflict with Commit

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

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

4.4.4 Write-Write Conflict with Rollback

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

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

4.4.5 Concurrent Progress of Read and Write Transactions

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

1. An ACID Transaction, T1, was started for a randomly selected O_KEY, L_KEY, and DELTA. T1 was suspended prior to ROLLBACK.
2. Another Transaction, T2, was started which did the following:
   For random values of PS_PARTKEY and PS_SUPPKEY, all columns of the PARTSUPP table for which PS_PARTKEY and PS_SUPPKEY are equal, are returned.
3. T2 completed.
4. T1 was allowed to COMMIT.
5. It was verified that appropriate rows in ORDERS, LINEITEM and HISTORY tables were changed.

4.4.6 Read-Only Query Conflict with Update Transaction

Demonstrate 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, executing Q1 against the qualification database, was started using a randomly selected DELTA.

2. An ACID Transaction T2, was started for a randomly selected O_KEY, L_KEY and DELTA.

3. T2 completed and appropriate rows in the ORDERS, LINEITEM and HISTORY tables had been changed.

4. Transaction T1 completed executing Q1.

4.5 Durability

The SUT 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.

4.5.1 Failure of a Durable Medium

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

The LUNs containing TPC-H data files and redo log files were pairs of physical disks mirrored by hardware (RAID1). Each TPC-H data file and redo log file was striped across all available LUNs by software (RAID0), i.e. each LUN contained pieces from all TPC-H data files and redo log files. During the durability test, a LUN (i.e. a mirrored pair of physical disks) was randomly chosen and one of its disks removed from the cabinet. The test continued uninterupted, using the remaining side of the mirror.

4.5.2 System Crash

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

The system crash and memory failure tests were combined. Power to the server was turned off by the "Partition forced Power-Off Instruction" to the System Control Facility processor (SCF). When power was restored, the system rebooted and the database was restarted. The durability success file and the HISTORY table were compared successfully.

4.5.3 Memory Failure

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

See section 4.5.2.

5 Clause 4 Scaling and Database

5.1 Ending Cardinality of Tables

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

Table	Rows
Orders	1,500,000,000
Lineitem	5,999,989,709
Customer	150,000,000
Part	200,000,000
Supplier	10,000,000
Partsupp	800,000,000
Nation	25
Region	5

5.2 Distribution of Tables and Logs Across Media

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

There were 64 FibreCAT S80 storage arrays with 10 physical disks each. For each of these S80, 5 LUNs were created from pairs of mirrored physical disks and shown to Solaris. Each S80 array was attached to the server by one Emulex LP9802 Fibre Channel Controller. The write caches of the S80 were disabled.

In all 320 LUNs were available and used for 2 Veritas Volume Manager disk groups. The first group consisted of 32 LUNs, taking 2 LUNs from each of 16 S80. This disk group was used for the dbgen flat files exclusively, not for the TPC-H database.

The second disk group consisted of the remaining 288 LUNs. For each of the Oracle data files and redo log files as listed in the database schema specification of Appendix B, a Veritas volume striped across all 288 LUNs was created. Appendix B shows the Veritas description (vxprint -th) for a volume as a sample.

5.3 Database partition/replication mapping

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

The database was not replicated.

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

5.4 RAID Feature

Implementations may use some form of RAID 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.

Table/Index	RAID type
tables	RAID 1+0
indexes	RAID 1+0
temp tablespace	RAID 1+0
log	RAID 1+0
System tablespace	RAID 1+0

5.5 Modifications to the DBGEN

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

The supplied DBGEN version 1.3.0 was used 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 2 hours 18 minutes 34 seconds.

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:

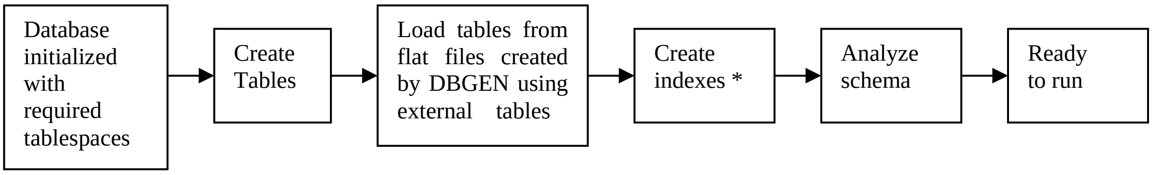
Disk Type	# Of Disks	Space Per Disk*	Sub-Total Disk Space**
internal	5	36.0 GB	167.65 GB
S80	640	73.0 GB	43,511.39 GB
		Total Space	**43,679.04 GB**
		Data Storage Ratio	**43.68**

*****Disk manufacturer definition of one GB is 10^9 byte ******In this calculation one GB is defined as 2^30 bytes

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 flat files all on the tested and priced configurations. Oracle created external tables using the files that were created by the DBGEN program.


Database initialized with required tablespaces
Create Tables
Load tables from flat files created by DBGEN using external tables
Create indexes *
Analyze schema
Ready to run

*Analyze index performed during index creation

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 adjustments for the size difference.

6 Clause 5 Performance Metrics and Execution Rules

6.1 System Activity Between Load and Performance Tests

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

Auditor requested queries were run against the database to verify the correctness of the load. All scripts and queries used are included in Appendix F.

6.2 Steps in the Power Test

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

The following steps were used to implement the power test:

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

6.3 Timing Intervals for Each Query and Refresh Functions

The timing intervals for each query and 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.

Eight streams were used for the throughput test.

6.5 Start and End Date/Times for 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 Quantity 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 Quantity Summary earlier in this document.

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

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

The start and finish times for each refresh function in the refresh stream are given in the Numerical Quantity Summary earlier in this document.

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

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

The timing intervals for each query and each refresh function for the throughput test are given in the Numerical Quantity 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, are given in the Numerical Quantity 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:

Run ID	QppH@1000GB	QthH@1000GB	QphH@1000GB
Run 1	42,734.1	28,815.0	35,091.1
Run 2	41,829.5	28,442.4	34,492.5
Difference	-2.16%	-1.31%	-1.74%

6.11 System Activity Between Performance Tests

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

There was no activity on the SUT between run1 and run2

7 Clause 6 SUT and Driver Implementation

7.1 Driver

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

The Power Test and Throughput Test are performed by a shell script called runTPCpt. QGEN is first called with a stream id of 0 to generate the QET for the Power Test. UF1 is then started by executing the runuf1.sh script. Query submission follows, with the qexecpl.c ISL program. The execution of the UF2 script runuf2.sh rounds out the Power Test execution.

Following the Power Test, QGEN is again called with the subsequent 8 stream ids to generate new QET for each Throughput Test. qexecpl.c is called simultaneously for all 8 streams to execute the queries as above.Then the update_stream.sh script is called to run all 8 update pairs to finish the throughput run.

7.2 Implementation-Specific Layer

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

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 5.2.9 is used, such use must be disclosed. Profile-directed optimization was not used.

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

8 Clause 7 Pricing

8.1 Hardware and Software Used

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

A detailed list of hardware and software used in the priced system is included in the pricing sheet in the Executive Summary.

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 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.

8.3 Availability Date

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

All hardware and software components will be available March 8, 2004.

9 Auditor's Information and Attestation Letter

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

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

Appendix A. Solaris 9 and Oracle Database 10g Parameters

This Appendix contains Oracle initialization parameters, environment variables and Solaris kernel parameters.

```
===========================================
Oracle 10g Parameters: init.ora
===========================================
audit_trail                        = FALSE
compatible                         =
10.0.0.0.0
control_files               =
(/tpch_df/dbs/ctl01.ctl
/tpch_df/dbs/ctl02.ctl)
cpu_count                          = 64
db_cache_size                      = 18g
db_block_checksum         = FALSE
db_block_size                      = 8192
db_file_multiblock_read_count      = 128
db_files                           = 1024
db_name                   = tpch
db_writer_processes                = 20
dml_locks                          = 120000
enqueue_resources                  = 50000

global_names                       = FALSE
java_pool_size                     = 0
large_pool_size           = 8g
log_buffer                         =
67108864
log_checkpoints_to_alert  = TRUE
max_dump_file_size                 = 50000
nls_date_format           = YYYY-MM-DD
open_cursors                       = 1024
optimizer_features_enable = 10.0.0.1
optimizer_index_cost_adj  = 40 #25
optimizer_mode                     = CHOOSE
parallel_execution_message_size = 16384
parallel_max_servers               = 1225
#1400 #96
parallel_min_servers               = 1225
#1400 #96
pga_aggregate_target               = 23g
#70g
processes                          = 2000
query_rewrite_enabled              = TRUE
replication_dependency_tracking = FALSE
sessions                           = 3000
shared_pool_size          = 8g
statistics_level          = BASIC
transactions                       = 512
undo_management                    = auto
```

```
==========================================
Oracle Environment Variables
==========================================
export KIT_DIR=$HOME/tpch
export SCHEMA_DIR=$KIT_DIR/bmc/schema
export PERL=/usr/bin/perl
export BUMPX_DIR=$KIT_DIR/bumpx
export BUMPX_OUT=$KIT_DIR/bumpx
export UTILS=$KIT_DIR/utils
export TEST_DB=/tmp
export QUAL_DB=$TEST_DB
export DBGEN=$KIT_DIR/dbgen
export ACID_DIR=$KIT_DIR/acid
export QEXEC=$KIT_DIR/utils
export QUERIES=$KIT_DIR/queries
export ANSWERS=$KIT_DIR/answers
export ANS2VAL=/tmp
export ACID_OUT=$QUAL_DB/acid_out
export DSS_CONFIG=$DBGEN
export DSS_QUERY=$KIT_DIR/queries
export DSS_PATH=$HOME
export MAINT=$KIT_DIR/maintenance
export CC=/usr/bin/cc
export FRAME=$KIT_DIR/frame
export
REGR_TEST=$KIT_DIR/internal/regression_test
export UPDATE_DOP_INS=96
export UPDATE_DOP_DEL=128
export SCALE_FACTOR=1000
############ FRAME STUFF
export FRAME_PATH=$KIT_DIR/frame
export ORACORE3INCL=$ORACLE_HOME/rdbms/demo
export
ORACORE3PUBL=$ORACLE_HOME/rdbms//public
export RDBMSPUBL=$ORACLE_HOME/rdbms/public
export
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}:${MA
INT}:${ACID_DIR}:${FRAME}/bin:${FRAME}/bin:
${REGR_TEST}:${PATH}
#
########### ENVIRONMENT VARIABLES
##################
export WORKLOAD=TPCH
export HOST=
export OPTLEVEL=
export GETOPT=-DSTDLIB_HAS_GETOPT
export PLATFORM=
export
INITORA=$KIT_DIR/schema/test_db/sf100.ora

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

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

```
DATABASE_USER=tpch/tpch
```

```
===========================================
Solaris Parameters: /etc/system
===========================================
* Begin FJSVssf (do not edit)
set ftrace_atboot = 1
set kmem_flags = 0x100
set kmem_lite_maxalign = 8192
* End FJSVssf (do not edit)
* END SSF
*******************************************
********************************
* Begin FJSVpnl (do not edit)
forceload:       drv/FJSVpanel
* End FJSVpnl (do not edit)
forceload:       drv/se
forceload:       drv/fjmse

forceload: drv/clone

set pcipsy:pci_stream_buf_enable = 0

set shmsys:shminfo_shmmax=0xffffffffffffffff
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=1024
set shmsys:shminfo_shmseg=500

set semsys:seminfo_semmap=8388608
set semsys:seminfo_semmni=4096
set semsys:seminfo_semmns=8388608
set semsys:seminfo_semmnu=4096
set semsys:seminfo_semmsl=2048
set semsys:seminfo_semume=2048
set semsys:seminfo_semopm=100
set semsys:seminfo_semvmx=32767

set msgsys:msginfo_msgmap=2048
set msgsys:msginfo_msgmax=8192
set msgsys:msginfo_msgmnb=16384
set msgsys:msginfo_msgssz=32
set msgsys:msginfo_msgtql=2048
set msgsys:msginfo_msgseg=32767

set maxpgio=131072
set maxphys=4194304
set bufhwm=8000
set segspt_minfree=16000
set tune_t_fsflushr=10
set autoup=1800
set memscrub_period_sec=172900

set seg_pwindow=28311552
set p_hashsize=131072
set seg_pmaxqlen=128
set segmap_percent=2

* vxvm_START (do not remove)
forceload: drv/vxdmp
forceload: drv/vxio
forceload: drv/vxspec
* vxvm_END (do not remove)
```

Appendix B. Programs and Scripts

```
===========================================
bumpx.pl
===========================================
#!/usr/local/bin/perl

$os = $ENV{'OS'};
if (($os cmp 'Windows_NT') != 0) { # os is UNIX
   $os = "unix";$nt = 0;$unix = 1;
} else {
   $os = "nt"; $nt = 1; $unix = 0;
}
$| = 1;
$verbose = 0;
if (($os cmp "unix")==0) {
   $defphases =
"dbcre,sctso,scuto,dbgen,dapop,anlyz,ixcre"
;
} else {
   $defphases =
"sdgen,shutd,start,dbgen,plcre,dbcre,sctso,
scuto,dapop,scuvo,anlyz,ixcre,chob";
}
$allbmtypes = "tpcd,wisc";
$bmtype = "tpcd" if !defined $bmtype;
$pdfile =  "$ENV{'BUMPX_DIR'}/param.txt";  #
This file contains the description of all
possible parameters.
while ($arg = shift(@ARGV)) {
   if ($arg !~ /-(i|o|t|p|d|a|s|h)/){
      $error =  "** Error: Bad argument to $0:
$arg\n";
      &usage;
   }
   if ($arg =~ /-h/) { &usage;ext(0);}
   $runsilent = 1 if ($arg =~ /-s/);
   $outfile = shift(@ARGV) if ($arg =~ /-o/);
   $bmtype = shift(@ARGV) if ($arg =~ /-t/);
   $phaselist = shift(@ARGV) if ($arg =~
/-p/);
   if ($arg =~ /-d/) {
         $defpar = shift(@ARGV);
         @keys = keys %params;
         while ($#keys >=0) {
            $key = pop(@keys);
            if (($defpar cmp "") ==0) {
                 print $key, "=",
$params{$key}, "\n";
            } else {
                 print $key, "=",
$params{$key}, "\n" if ($key =~ /$defpar/);
            }
         }
         exit(0);
   }
}
$outfile = "$ENV{'BUMPX_DIR'}/bumpx.dat"
if !defined $outfile;
if ($nt) {
   $listdir = $filedir."list/";
   if (!-e $listfile) {
        system ("mkdir $listdir");
   }
}
if (($os cmp "nt") == 0) {## NT Port (Use
tmpfile to buffer
   $tmpfile = "tmp.txt";      ## commands and
nruntpb to synchronize them)
   $tmpfile = $filedir.$tmpfile;
   $nruntpb =  "nruntpb.exe";
} ## NT End
if (!-e $outfile) {
   $error = "** Error: -o file, $outfile, does
not exist\n";
   &usage;
}
$phaselist = $defphases if !defined
$phaselist;
@phases = split(/,/, $phaselist);
## NT Port (Use tmpfile to buffer commands for
nruntpb)
open (TMPFILE, ">$tmpfile") if ( (($os cmp
"nt") == 0));
## NT End
&doexecute;
## NT Port
close(TMPFILE) if ( (($os cmp "nt") == 0));
## NT End
exit(0);

sub doexecute { # First, do preprocessing stuff
   print "Execution pass begun." if $verbose;
   open (INFILE, $outfile);
  WLOOP1:
   while ($line = <INFILE>)
   {
      study $line;
      next WLOOP1 if $line =~ /^\s*\#/;
      next WLOOP1 if $line =~ /^\s*\n/;
      if ($line =~ /^%b-preproc/)
      {
         $insection = 1;
         next WLOOP1;
      }
      next WLOOP1 if ($insection != 1);
      if ($line =~ /^%e-preproc/)
      {
         $insection = 0;
         $commands{$shortcmd} = $longcmd if
defined $shortcmd;
         last WLOOP1;
      }
      if ($line =~ /^\*/)
      {
         $commands{$shortcmd} = $longcmd if
defined $shortcmd;
         $line =~ /^(\*.*\S+)\s*\n$/;
         $shortcmd = $1;
         $longcmd = "";
         next WLOOP1;
      }
      if ($line =~ /^\\/)
      {
         $line =~ /\\(.*\n)/;
         $longcmd = $longcmd . $1;
         next WLOOP1;
      }
      print "Illegal entry in preproc
stage:\n   $line";
   }
   close (INFILE);
```

```
   $execctr = 0;
   foreach $phase (@phases)
   {
        $phase_cmd_num = 0;
        print "\n  Executing phase
\'$phase\'" if $verbose;
        $bg = 0;
      open (INFILE, $outfile);
     WLOOP2:
      while ($line = <INFILE>)
      {
         study $line;
         next WLOOP2 if $line =~ /^\s*\#/;
         next WLOOP2 if $line =~ /^\s*\n/;
         if ($line =~ /^\*ignon/)
         {
            $ignon = 1;
            next WLOOP2;
         }
         if ($line =~ /^\*ignoff/)
         {
            $ignon = 0;
            next WLOOP2;
         }
         next WLOOP2 if ($ignon == 1);
         if ($line =~ /^%b-$phase/)
         {
            $insection = 1;
            $execcmd = "";
            next WLOOP2;
         }
         next WLOOP2 if ($insection != 1);
         if ($line =~ /^%e-$phase/)
         {
            $insection = 0;
            &execute ($execcmd);
            last WLOOP2;
         }
         if ($line =~ /^\*(.*)/)
         {
            &execute ($execcmd);
            if (($1 =~ /bgo/) || ($1 =~
/wait/) || ($1 =~ /ignore/))
            {
               $execcmd = $line;
               next WLOOP2;
            }
            $line =~ /^(\*.*\S+)\s*\n$/;
            $execcmd = $commands{$1};
            next WLOOP2;
         }
         if ($line =~ /^\{(.*)\}/)
         {
            $insert = "";
            $insert = $1;
            $execcmd =~ s/\{\}/$insert/;
            next WLOOP2;
         }
         if ($line =~ /^\{(.*)$/)
         {
            $insubsection = 1;
            $insert = "";
            $insert = $1;
            next WLOOP2;
         }
         if ($line =~ /^(.*)\}/)
         {
            $insubsection = 0;
            $insert = $insert . $1;
                if (($os cmp "nt") == 0){ ##
NT Port (Ignore '\n')
```

```
                    $insert =~ /(.*)\n$/s;
                    $insert = $1;
                 } ## NT End
             $execcmd =~ s/\{\}/$insert/;
             next WLOOP2;
          }
          $insert = $insert . $line if
($insubsection == 1);
       }
       close (INFILE);
   }
   print "\nExecution pass complete.\n" if
$verbose;
}

sub execute
{
   $cmd = shift(@_);
   if ($cmd)
   {
      return if ($cmd =~ /^\*ignore/);
      if ($cmd =~ /^\*bgon=(.*)/)
      {
         $bgmax = $1;
         $bg = 1;
         $bgrun = 0;
         return;
      }
      if ($cmd =~ /^\*bgoff/)
      {
         $bg = 0;
         return;
      }

        if ($cmd =~ /^\*time=(.*)/)  ##NT
only
      {
           print $1 . "\n";
           print localtime(time) . "\n";
         return;
      }
      if ($cmd =~ /^copy (.*)/) ## NT only
      {
           system($cmd);
         ## Quit if failed
         if ($?) {
                print "system copy command
failed:\n$cmd\nreason: $? ($!)\n";
              exit(-1);
           }
         return;
      }
        if ($cmd =~ /^del (.*)/) ## NT only
      {
           system($cmd);
         ## Quit if failed
         if ($?) {
                print "system del command
failed:\n$cmd\nreason: $? ($!)\n";
              exit(-1);
           }
         return;
      }

        if ($cmd =~ /^\*wait/) ## This deals
with main differences between NT and UNIX
      {
           if (($os cmp "unix") == 0)
           {
                while ($fpid =
shift(@wpids))
                {
                   waitpid($fpid, 0);
                }
           }
           else
           {    ## NT Port (Start background
tasks if any. nruntpb will wait until all tasks
are done)
                if ($bgrun >= 1)
                {
                   close(TMPFILE);
                   system("cat $tmpfile >>
$listdir$phase.lst");
                   system("vi $tmpfile")
if $debug;
                   system("$nruntpb -p <
$tmpfile") if !$debug;
                   if ($?)
                   {
                        print "system
command failed:\n$nruntpb < $tmpfile\n";
                        print "reason: $?
($!)\n";
                        print "Please
check the contents in the input file.\n";
                        exit(-1);
                   }
                   open(TMPFILE,
">$tmpfile");
                }
           }
         $bgrun = 0;
         return;
      }
      if ($cmd =~ /(s|g)etenv/)
      {
           @lines = split(/\n/, $cmd);
         $cmd = "";
         foreach $line (@lines)
         {
            while (1)
            {
               last if ($line !~ /getenv/);
               $line =~
/(.*)\*getenv\(([^(\)|\*)]*)\)(.*)/;
               $line = $1 . $ENV{$2} . $3;
            }
            if ($line =~ /jojo/) #we do not
want to use this for now
            {
               $line =~
/setenv\s+(\S+)\s+(\S+)/;
               $ENV{$1} = $2;
            }
            else
            {
               $cmd = $cmd . $line. "\n";
            }
         }
      }
      return if ($cmd !~ /\S+/); # return if
nothing left to execute
         $execctr++;
      $ENV{'BUMPX_CTR'} = $$.'-'.$execctr;
         if (($os cmp "unix") == 0)
         {
           if ($bg == 1)
           {
                print "." if $verbose;
                $fpid = fork;
                if ($fpid == 0)
                {
                   exec ($cmd);
```

```
                     print "exec\'d command
failed:\n$cmd\nreason: $!\n";
                     exit(-1);
                  }
                  unshift (@wpids, $fpid);
                  $bgrun = $bgrun + 1;
                  &execute ("*wait") if
(($bgrun >= $bgmax) && ($bgmax >= 0));
            }
            else
            {
                  system ($cmd);
                  print "system\'d command
failed:\n$cmd\nreason: $? ($!)\n" if $?;
            }
         }
         else ## NT support
         {
       ## NT Port (Submit background tasks if
there are bgmax of them, otherwise write to
tmpfile)
            if ($bg == 1)
            {
                  print "." if $verbose;
                  if ($bgrun < $bgmax)
                  {
                     $cmd =~
s/phase\#.lst/$listdir$phase\_$phase_cmd_nu
m.lst/;
                     ++$phase_cmd_num;
                     print TMPFILE $cmd;
                     $bgrun = $bgrun + 1;
                  }
                  else
                  {
                     close(TMPFILE);
                     system("cat $tmpfile >>
$listdir$phase.lst");
                     system("$nruntpb -p <
$tmpfile");
                     if ($?) {
                          print "system
command failed:\n$nruntpb < $tmpfile\nreason:
$? ($!)\n";
                          print "Please
check the contents in the input file.\n";
                          exit(-1);
                     }
                     open(TMPFILE,
">$tmpfile");
                     $cmd =~
s/phase\#.lst/$listdir$phase\_$phase_cmd_nu
m.lst/;
                     ++$phase_cmd_num;
                     print TMPFILE $cmd;
                     $bgrun = 1;
                  }
            }
            else
            {
                  $cmd =~
s/phase\#.lst/$listdir$phase\_$phase_cmd_nu
m.lst/;
                  ++$phase_cmd_num;
                  print TMPFILE $cmd;
                  close(TMPFILE);
                  system("cat $tmpfile >>
$listdir$phase.lst");
                  system ("sh $tmpfile");
                  if ($?) {
```

```
                     print "system\'d
command failed:\nsh $tmpfile\nreason: $?
($!)\n";
                     print "Please check the
contents in the shell script.\n";
                     exit(-1);
                  }
                  open(TMPFILE,
">$tmpfile");
            }
         }  ## NT support End
   }
}

sub usage
{
   print "Usage:\n";
   print "This is a lite version of bumpx.pl.
It can only be used to execute a .dat file\n";
   print " $0 [-o outfile] [-p phaselist] [-t
type]\n";
   print "   -o : intermediary file to be
created and/or used\n";
   print "      defaults to bumpx.dat in
\$BUMPX_DIR or \$CWD\n";
   print "   -p : list of phases to
create/execute\n";
   print "      phaselist is a comma
separated list of phases in order\n";
   print "      possible phases are:\n";
   print "        sdgen = seed file
generation\n";
   print "        dbgen = data flat file
generation\n";
   print "        plcre = NT raw partition and
links creation\n";
   print "        dbcre = database
creation\n";
   print "        shutd = shutdown database
(on all instances)\n";
   print "        start = startup database
(on all instances)\n";
   print "        sccre = schema creation\n";
   print "        sctso = schema creation
(tablespaces only)\n";
   print "        scuto = schema creation
(user and tables only)\n";
   print "        scuvo = schema creation
(views only)\n";
   print "        dapop = data population\n";
   print "        ixcre = index creation
(including constraints)\n";
   print "        anlyz = analyze objects\n";
   print "        chob  = change parameters
of objects\n";
   print "        expln = create explain
plans\n";
   print "        query = run and time
queries\n";
   print "      defaults to $defphases\n";
   print "   -t : type of benchmark\n";
   print "      enables benchmark-specific
defaults\n";
   print "      current possibilties are:
$allbmtypes\n";
   print "      defaults to tpcd\n";
   print "   -s : run silent (no parameter
checking is done)\n";
   print "\n";
   print "Examples:\n";
   print " $0 -p dapop\n";
```

```
    print "   Executes data population phase of
intermediary file bumpx.dat.\n";
    print "\n";
    print "$error\n";
    exit(-1);
}
```

```
===========================================
```

1tb.dat

```
===========================================
###########################################
#############################
# preprocessing-like directives
%b-preproc
*sql
\sqlplus -s /NOLOG <<!
\set echo on;
\set termout on;
\spool phase#.lst;
\connect / as sysdba;
\select to_char(sysdate, 'MM-DD-YYYY
HH24:MI:SS') now from dual;
\{}
\select to_char(sysdate, 'MM-DD-YYYY
HH24:MI:SS') now from dual;
\exit;
\!

*load1
\sqlldr {}

*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-dbcre
*bgon=1
###########################################
##################################
# Database Creation Phase
*sql
{
shutdown abort;
}
*wait
# creating database
*sql
{
startup pfile=
/export/home/oracle/tpch/admin/init_build.o
ra nomount;
create database
  controlfile reuse
  logfile '/tpch_df/log_1' size 31000m reuse,
         '/tpch_df/log_2' size 31000m reuse
  datafile '/tpch_df/sys_1' size 3300m reuse
  sysaux datafile '/tpch_df/sys_2' size 1000m
reuse
  undo tablespace ts_undo1
  datafile '/tpch_df/undo_1' size 30570m reuse
  maxdatafiles 5000
  maxinstances 1
;
}
*wait
# creating extra logfile threads for rac 1 nodes
*sql
{
}
*wait
# building data dictionary
*sql
{
set termout off
set echo off
spool
/export/home/oracle/tpch/log/data_dict.out
@?/rdbms/admin/catalog.sql;
@?/rdbms/admin/catparr.sql;
@?/rdbms/admin/catproc.sql;
connect system/manager
@?/rdbms/admin/utlxplan.sql;
@/ora10100/sqlplus/admin/pupbld.sql;
}
*wait
*sql
{
shutdown;
}
*wait
*sql
{
startup
pfile=/export/home/oracle/tpch/admin/init_b
uild.ora
}
*wait
*bgoff
%e-dbcre
%b-sctso
*bgon=128
###########################################
##################################
# Schema Creation Phase - datafiles only (no
tables or users)
# creating data tablespaces, datafiles
# creating tpch's ts_s tablespace
*sql
{
drop tablespace ts_s including contents;
create tablespace ts_s
datafile '/tpch_df/s_1' size 1900m reuse
extent management local
autoallocate
 nologging
;
}
# creating tpch's ts_c tablespace
*sql
{
drop tablespace ts_c including contents;
create tablespace ts_c
datafile '/tpch_df/c_1' size 24860m reuse
extent management local
autoallocate
 nologging
;
}
# creating tpch's ts_ps tablespace
*sql
{
drop tablespace ts_ps including contents;
create tablespace ts_ps
datafile '/tpch_df/ps_1' size 29010m reuse
extent management local
autoallocate
 nologging
;
}
# creating tpch's ts_p tablespace
*sql
{
drop tablespace ts_p including contents;
create tablespace ts_p
datafile '/tpch_df/p_1' size 27795m reuse
extent management local
autoallocate
 nologging
;
}
# creating tpch's ts_def tablespace
*sql
{
drop tablespace ts_def including contents;
create tablespace ts_def
datafile '/tpch_df/def_1' size 3300m reuse
;
}
# creating tpch's ts_o1 tablespace
*sql
{
drop tablespace ts_o1 including contents;
create tablespace ts_o1
datafile '/tpch_df/o_1' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o2 tablespace
*sql
{
drop tablespace ts_o2 including contents;
create tablespace ts_o2
datafile '/tpch_df/o_2' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o3 tablespace
*sql
{
drop tablespace ts_o3 including contents;
create tablespace ts_o3
datafile '/tpch_df/o_3' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o4 tablespace
*sql
{
drop tablespace ts_o4 including contents;
create tablespace ts_o4
datafile '/tpch_df/o_4' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o5 tablespace
*sql
{
drop tablespace ts_o5 including contents;
create tablespace ts_o5
datafile '/tpch_df/o_5' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o6 tablespace
*sql
{
drop tablespace ts_o6 including contents;
create tablespace ts_o6
datafile '/tpch_df/o_6' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o7 tablespace
*sql
{
drop tablespace ts_o7 including contents;
create tablespace ts_o7
datafile '/tpch_df/o_7' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o8 tablespace
*sql
{
drop tablespace ts_o8 including contents;
create tablespace ts_o8
datafile '/tpch_df/o_8' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o9 tablespace
*sql
{
drop tablespace ts_o9 including contents;
create tablespace ts_o9
datafile '/tpch_df/o_9' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o10 tablespace
*sql
{
drop tablespace ts_o10 including contents;
create tablespace ts_o10
datafile '/tpch_df/o_10' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o11 tablespace
*sql
{
drop tablespace ts_o11 including contents;
create tablespace ts_o11
datafile '/tpch_df/o_11' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o12 tablespace
*sql
{
drop tablespace ts_o12 including contents;
create tablespace ts_o12
datafile '/tpch_df/o_12' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o13 tablespace
```

```
*sql
{
drop tablespace ts_o13 including contents;
create tablespace ts_o13
datafile '/tpch_df/o_13' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o14 tablespace
*sql
{
drop tablespace ts_o14 including contents;
create tablespace ts_o14
datafile '/tpch_df/o_14' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o15 tablespace
*sql
{
drop tablespace ts_o15 including contents;
create tablespace ts_o15
datafile '/tpch_df/o_15' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o16 tablespace
*sql
{
drop tablespace ts_o16 including contents;
create tablespace ts_o16
datafile '/tpch_df/o_16' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o17 tablespace
*sql
{
drop tablespace ts_o17 including contents;
create tablespace ts_o17
datafile '/tpch_df/o_17' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o18 tablespace
*sql
{
drop tablespace ts_o18 including contents;
create tablespace ts_o18
datafile '/tpch_df/o_18' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o19 tablespace
*sql
{
drop tablespace ts_o19 including contents;
create tablespace ts_o19
datafile '/tpch_df/o_19' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o20 tablespace
*sql
{
drop tablespace ts_o20 including contents;
create tablespace ts_o20
datafile '/tpch_df/o_20' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o21 tablespace
*sql
{
drop tablespace ts_o21 including contents;
create tablespace ts_o21
datafile '/tpch_df/o_21' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o22 tablespace
*sql
{
drop tablespace ts_o22 including contents;
create tablespace ts_o22
datafile '/tpch_df/o_22' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o23 tablespace
*sql
{
drop tablespace ts_o23 including contents;
create tablespace ts_o23
datafile '/tpch_df/o_23' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o24 tablespace
*sql
{
drop tablespace ts_o24 including contents;
create tablespace ts_o24
datafile '/tpch_df/o_24' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o25 tablespace
*sql
{
drop tablespace ts_o25 including contents;
create tablespace ts_o25
datafile '/tpch_df/o_25' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o26 tablespace
*sql
{
drop tablespace ts_o26 including contents;
create tablespace ts_o26
datafile '/tpch_df/o_26' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o27 tablespace
*sql
{
drop tablespace ts_o27 including contents;
create tablespace ts_o27
```

```
datafile '/tpch_df/o_27' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o28 tablespace
*sql
{
drop tablespace ts_o28 including contents;
create tablespace ts_o28
datafile '/tpch_df/o_28' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o29 tablespace
*sql
{
drop tablespace ts_o29 including contents;
create tablespace ts_o29
datafile '/tpch_df/o_29' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o30 tablespace
*sql
{
drop tablespace ts_o30 including contents;
create tablespace ts_o30
datafile '/tpch_df/o_30' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o31 tablespace
*sql
{
drop tablespace ts_o31 including contents;
create tablespace ts_o31
datafile '/tpch_df/o_31' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o32 tablespace
*sql
{
drop tablespace ts_o32 including contents;
create tablespace ts_o32
datafile '/tpch_df/o_32' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o33 tablespace
*sql
{
drop tablespace ts_o33 including contents;
create tablespace ts_o33
datafile '/tpch_df/o_33' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o34 tablespace
*sql
{
drop tablespace ts_o34 including contents;
create tablespace ts_o34
datafile '/tpch_df/o_34' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o35 tablespace
*sql
{
drop tablespace ts_o35 including contents;
create tablespace ts_o35
datafile '/tpch_df/o_35' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o36 tablespace
*sql
{
drop tablespace ts_o36 including contents;
create tablespace ts_o36
datafile '/tpch_df/o_36' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o37 tablespace
*sql
{
drop tablespace ts_o37 including contents;
create tablespace ts_o37
datafile '/tpch_df/o_37' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o38 tablespace
*sql
{
drop tablespace ts_o38 including contents;
create tablespace ts_o38
datafile '/tpch_df/o_38' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o39 tablespace
*sql
{
drop tablespace ts_o39 including contents;
create tablespace ts_o39
datafile '/tpch_df/o_39' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o40 tablespace
*sql
{
drop tablespace ts_o40 including contents;
create tablespace ts_o40
datafile '/tpch_df/o_40' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o41 tablespace
*sql
{
drop tablespace ts_o41 including contents;
create tablespace ts_o41
datafile '/tpch_df/o_41' size 3100m reuse
extent management local
autoallocate
;
```

```
}
# creating tpch's ts_o42 tablespace
*sql
{
drop tablespace ts_o42 including contents;
create tablespace ts_o42
datafile '/tpch_df/o_42' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o43 tablespace
*sql
{
drop tablespace ts_o43 including contents;
create tablespace ts_o43
datafile '/tpch_df/o_43' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o44 tablespace
*sql
{
drop tablespace ts_o44 including contents;
create tablespace ts_o44
datafile '/tpch_df/o_44' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o45 tablespace
*sql
{
drop tablespace ts_o45 including contents;
create tablespace ts_o45
datafile '/tpch_df/o_45' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o46 tablespace
*sql
{
drop tablespace ts_o46 including contents;
create tablespace ts_o46
datafile '/tpch_df/o_46' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o47 tablespace
*sql
{
drop tablespace ts_o47 including contents;
create tablespace ts_o47
datafile '/tpch_df/o_47' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o48 tablespace
*sql
{
drop tablespace ts_o48 including contents;
create tablespace ts_o48
datafile '/tpch_df/o_48' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o49 tablespace
*sql
{
drop tablespace ts_o49 including contents;
create tablespace ts_o49
datafile '/tpch_df/o_49' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o50 tablespace
*sql
{
drop tablespace ts_o50 including contents;
create tablespace ts_o50
datafile '/tpch_df/o_50' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o51 tablespace
*sql
{
drop tablespace ts_o51 including contents;
create tablespace ts_o51
datafile '/tpch_df/o_51' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o52 tablespace
*sql
{
drop tablespace ts_o52 including contents;
create tablespace ts_o52
datafile '/tpch_df/o_52' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o53 tablespace
*sql
{
drop tablespace ts_o53 including contents;
create tablespace ts_o53
datafile '/tpch_df/o_53' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o54 tablespace
*sql
{
drop tablespace ts_o54 including contents;
create tablespace ts_o54
datafile '/tpch_df/o_54' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o55 tablespace
*sql
{
drop tablespace ts_o55 including contents;
create tablespace ts_o55
datafile '/tpch_df/o_55' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o56 tablespace
*sql
{
```

```
drop tablespace ts_o56 including contents;
create tablespace ts_o56
datafile '/tpch_df/o_56' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o57 tablespace
*sql
{
drop tablespace ts_o57 including contents;
create tablespace ts_o57
datafile '/tpch_df/o_57' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o58 tablespace
*sql
{
drop tablespace ts_o58 including contents;
create tablespace ts_o58
datafile '/tpch_df/o_58' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o59 tablespace
*sql
{
drop tablespace ts_o59 including contents;
create tablespace ts_o59
datafile '/tpch_df/o_59' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o60 tablespace
*sql
{
drop tablespace ts_o60 including contents;
create tablespace ts_o60
datafile '/tpch_df/o_60' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o61 tablespace
*sql
{
drop tablespace ts_o61 including contents;
create tablespace ts_o61
datafile '/tpch_df/o_61' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o62 tablespace
*sql
{
drop tablespace ts_o62 including contents;
create tablespace ts_o62
datafile '/tpch_df/o_62' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o63 tablespace
*sql
{
drop tablespace ts_o63 including contents;
create tablespace ts_o63
datafile '/tpch_df/o_63' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o64 tablespace
*sql
{
drop tablespace ts_o64 including contents;
create tablespace ts_o64
datafile '/tpch_df/o_64' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o65 tablespace
*sql
{
drop tablespace ts_o65 including contents;
create tablespace ts_o65
datafile '/tpch_df/o_65' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o66 tablespace
*sql
{
drop tablespace ts_o66 including contents;
create tablespace ts_o66
datafile '/tpch_df/o_66' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o67 tablespace
*sql
{
drop tablespace ts_o67 including contents;
create tablespace ts_o67
datafile '/tpch_df/o_67' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o68 tablespace
*sql
{
drop tablespace ts_o68 including contents;
create tablespace ts_o68
datafile '/tpch_df/o_68' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o69 tablespace
*sql
{
drop tablespace ts_o69 including contents;
create tablespace ts_o69
datafile '/tpch_df/o_69' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o70 tablespace
*sql
{
drop tablespace ts_o70 including contents;
create tablespace ts_o70
datafile '/tpch_df/o_70' size 3100m reuse
extent management local
```

```
autoallocate
;
}
# creating tpch's ts_o71 tablespace
*sql
{
drop tablespace ts_o71 including contents;
create tablespace ts_o71
datafile '/tpch_df/o_71' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o72 tablespace
*sql
{
drop tablespace ts_o72 including contents;
create tablespace ts_o72
datafile '/tpch_df/o_72' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o73 tablespace
*sql
{
drop tablespace ts_o73 including contents;
create tablespace ts_o73
datafile '/tpch_df/o_73' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o74 tablespace
*sql
{
drop tablespace ts_o74 including contents;
create tablespace ts_o74
datafile '/tpch_df/o_74' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o75 tablespace
*sql
{
drop tablespace ts_o75 including contents;
create tablespace ts_o75
datafile '/tpch_df/o_75' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o76 tablespace
*sql
{
drop tablespace ts_o76 including contents;
create tablespace ts_o76
datafile '/tpch_df/o_76' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o77 tablespace
*sql
{
drop tablespace ts_o77 including contents;
create tablespace ts_o77
datafile '/tpch_df/o_77' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o78 tablespace
*sql
{
drop tablespace ts_o78 including contents;
create tablespace ts_o78
datafile '/tpch_df/o_78' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o79 tablespace
*sql
{
drop tablespace ts_o79 including contents;
create tablespace ts_o79
datafile '/tpch_df/o_79' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o80 tablespace
*sql
{
drop tablespace ts_o80 including contents;
create tablespace ts_o80
datafile '/tpch_df/o_80' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o81 tablespace
*sql
{
drop tablespace ts_o81 including contents;
create tablespace ts_o81
datafile '/tpch_df/o_81' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o82 tablespace
*sql
{
drop tablespace ts_o82 including contents;
create tablespace ts_o82
datafile '/tpch_df/o_82' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o83 tablespace
*sql
{
drop tablespace ts_o83 including contents;
create tablespace ts_o83
datafile '/tpch_df/o_83' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_o84 tablespace
*sql
{
drop tablespace ts_o84 including contents;
create tablespace ts_o84
datafile '/tpch_df/o_84' size 3100m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l1 tablespace
```

```
*sql
{
drop tablespace ts_l1 including contents;
create tablespace ts_l1
datafile '/tpch_df/l_1' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l2 tablespace
*sql
{
drop tablespace ts_l2 including contents;
create tablespace ts_l2
datafile '/tpch_df/l_2' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l3 tablespace
*sql
{
drop tablespace ts_l3 including contents;
create tablespace ts_l3
datafile '/tpch_df/l_3' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l4 tablespace
*sql
{
drop tablespace ts_l4 including contents;
create tablespace ts_l4
datafile '/tpch_df/l_4' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l5 tablespace
*sql
{
drop tablespace ts_l5 including contents;
create tablespace ts_l5
datafile '/tpch_df/l_5' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l6 tablespace
*sql
{
drop tablespace ts_l6 including contents;
create tablespace ts_l6
datafile '/tpch_df/l_6' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l7 tablespace
*sql
{
drop tablespace ts_l7 including contents;
create tablespace ts_l7
datafile '/tpch_df/l_7' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l8 tablespace
*sql
{
drop tablespace ts_l8 including contents;
create tablespace ts_l8
datafile '/tpch_df/l_8' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l9 tablespace
*sql
{
drop tablespace ts_l9 including contents;
create tablespace ts_l9
datafile '/tpch_df/l_9' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l10 tablespace
*sql
{
drop tablespace ts_l10 including contents;
create tablespace ts_l10
datafile '/tpch_df/l_10' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l11 tablespace
*sql
{
drop tablespace ts_l11 including contents;
create tablespace ts_l11
datafile '/tpch_df/l_11' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l12 tablespace
*sql
{
drop tablespace ts_l12 including contents;
create tablespace ts_l12
datafile '/tpch_df/l_12' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l13 tablespace
*sql
{
drop tablespace ts_l13 including contents;
create tablespace ts_l13
datafile '/tpch_df/l_13' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l14 tablespace
*sql
{
drop tablespace ts_l14 including contents;
create tablespace ts_l14
datafile '/tpch_df/l_14' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l15 tablespace
*sql
{
drop tablespace ts_l15 including contents;
create tablespace ts_l15
```

```
datafile '/tpch_df/l_15' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l16 tablespace
*sql
{
drop tablespace ts_l16 including contents;
create tablespace ts_l16
datafile '/tpch_df/l_16' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l17 tablespace
*sql
{
drop tablespace ts_l17 including contents;
create tablespace ts_l17
datafile '/tpch_df/l_17' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l18 tablespace
*sql
{
drop tablespace ts_l18 including contents;
create tablespace ts_l18
datafile '/tpch_df/l_18' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l19 tablespace
*sql
{
drop tablespace ts_l19 including contents;
create tablespace ts_l19
datafile '/tpch_df/l_19' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l20 tablespace
*sql
{
drop tablespace ts_l20 including contents;
create tablespace ts_l20
datafile '/tpch_df/l_20' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l21 tablespace
*sql
{
drop tablespace ts_l21 including contents;
create tablespace ts_l21
datafile '/tpch_df/l_21' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l22 tablespace
*sql
{
drop tablespace ts_l22 including contents;
create tablespace ts_l22
datafile '/tpch_df/l_22' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l23 tablespace
*sql
{
drop tablespace ts_l23 including contents;
create tablespace ts_l23
datafile '/tpch_df/l_23' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l24 tablespace
*sql
{
drop tablespace ts_l24 including contents;
create tablespace ts_l24
datafile '/tpch_df/l_24' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l25 tablespace
*sql
{
drop tablespace ts_l25 including contents;
create tablespace ts_l25
datafile '/tpch_df/l_25' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l26 tablespace
*sql
{
drop tablespace ts_l26 including contents;
create tablespace ts_l26
datafile '/tpch_df/l_26' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l27 tablespace
*sql
{
drop tablespace ts_l27 including contents;
create tablespace ts_l27
datafile '/tpch_df/l_27' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l28 tablespace
*sql
{
drop tablespace ts_l28 including contents;
create tablespace ts_l28
datafile '/tpch_df/l_28' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l29 tablespace
*sql
{
drop tablespace ts_l29 including contents;
create tablespace ts_l29
datafile '/tpch_df/l_29' size 14405m reuse
extent management local
autoallocate
;
```

```
}
# creating tpch's ts_l30 tablespace
*sql
{
drop tablespace ts_l30 including contents;
create tablespace ts_l30
datafile '/tpch_df/l_30' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l31 tablespace
*sql
{
drop tablespace ts_l31 including contents;
create tablespace ts_l31
datafile '/tpch_df/l_31' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l32 tablespace
*sql
{
drop tablespace ts_l32 including contents;
create tablespace ts_l32
datafile '/tpch_df/l_32' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l33 tablespace
*sql
{
drop tablespace ts_l33 including contents;
create tablespace ts_l33
datafile '/tpch_df/l_33' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l34 tablespace
*sql
{
drop tablespace ts_l34 including contents;
create tablespace ts_l34
datafile '/tpch_df/l_34' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l35 tablespace
*sql
{
drop tablespace ts_l35 including contents;
create tablespace ts_l35
datafile '/tpch_df/l_35' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l36 tablespace
*sql
{
drop tablespace ts_l36 including contents;
create tablespace ts_l36
datafile '/tpch_df/l_36' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l37 tablespace
*sql
{
drop tablespace ts_l37 including contents;
create tablespace ts_l37
datafile '/tpch_df/l_37' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l38 tablespace
*sql
{
drop tablespace ts_l38 including contents;
create tablespace ts_l38
datafile '/tpch_df/l_38' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l39 tablespace
*sql
{
drop tablespace ts_l39 including contents;
create tablespace ts_l39
datafile '/tpch_df/l_39' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l40 tablespace
*sql
{
drop tablespace ts_l40 including contents;
create tablespace ts_l40
datafile '/tpch_df/l_40' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l41 tablespace
*sql
{
drop tablespace ts_l41 including contents;
create tablespace ts_l41
datafile '/tpch_df/l_41' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l42 tablespace
*sql
{
drop tablespace ts_l42 including contents;
create tablespace ts_l42
datafile '/tpch_df/l_42' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l43 tablespace
*sql
{
drop tablespace ts_l43 including contents;
create tablespace ts_l43
datafile '/tpch_df/l_43' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l44 tablespace
*sql
{
```

```
drop tablespace ts_l44 including contents;
create tablespace ts_l44
datafile '/tpch_df/l_44' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l45 tablespace
*sql
{
drop tablespace ts_l45 including contents;
create tablespace ts_l45
datafile '/tpch_df/l_45' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l46 tablespace
*sql
{
drop tablespace ts_l46 including contents;
create tablespace ts_l46
datafile '/tpch_df/l_46' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l47 tablespace
*sql
{
drop tablespace ts_l47 including contents;
create tablespace ts_l47
datafile '/tpch_df/l_47' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l48 tablespace
*sql
{
drop tablespace ts_l48 including contents;
create tablespace ts_l48
datafile '/tpch_df/l_48' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l49 tablespace
*sql
{
drop tablespace ts_l49 including contents;
create tablespace ts_l49
datafile '/tpch_df/l_49' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l50 tablespace
*sql
{
drop tablespace ts_l50 including contents;
create tablespace ts_l50
datafile '/tpch_df/l_50' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l51 tablespace
*sql
{
drop tablespace ts_l51 including contents;
create tablespace ts_l51
datafile '/tpch_df/l_51' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l52 tablespace
*sql
{
drop tablespace ts_l52 including contents;
create tablespace ts_l52
datafile '/tpch_df/l_52' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l53 tablespace
*sql
{
drop tablespace ts_l53 including contents;
create tablespace ts_l53
datafile '/tpch_df/l_53' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l54 tablespace
*sql
{
drop tablespace ts_l54 including contents;
create tablespace ts_l54
datafile '/tpch_df/l_54' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l55 tablespace
*sql
{
drop tablespace ts_l55 including contents;
create tablespace ts_l55
datafile '/tpch_df/l_55' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l56 tablespace
*sql
{
drop tablespace ts_l56 including contents;
create tablespace ts_l56
datafile '/tpch_df/l_56' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l57 tablespace
*sql
{
drop tablespace ts_l57 including contents;
create tablespace ts_l57
datafile '/tpch_df/l_57' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l58 tablespace
*sql
{
drop tablespace ts_l58 including contents;
create tablespace ts_l58
datafile '/tpch_df/l_58' size 14405m reuse
extent management local
```

```
autoallocate
;
}
# creating tpch's ts_l59 tablespace
*sql
{
drop tablespace ts_l59 including contents;
create tablespace ts_l59
datafile '/tpch_df/l_59' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l60 tablespace
*sql
{
drop tablespace ts_l60 including contents;
create tablespace ts_l60
datafile '/tpch_df/l_60' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l61 tablespace
*sql
{
drop tablespace ts_l61 including contents;
create tablespace ts_l61
datafile '/tpch_df/l_61' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l62 tablespace
*sql
{
drop tablespace ts_l62 including contents;
create tablespace ts_l62
datafile '/tpch_df/l_62' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l63 tablespace
*sql
{
drop tablespace ts_l63 including contents;
create tablespace ts_l63
datafile '/tpch_df/l_63' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l64 tablespace
*sql
{
drop tablespace ts_l64 including contents;
create tablespace ts_l64
datafile '/tpch_df/l_64' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l65 tablespace
*sql
{
drop tablespace ts_l65 including contents;
create tablespace ts_l65
datafile '/tpch_df/l_65' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l66 tablespace
*sql
{
drop tablespace ts_l66 including contents;
create tablespace ts_l66
datafile '/tpch_df/l_66' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l67 tablespace
*sql
{
drop tablespace ts_l67 including contents;
create tablespace ts_l67
datafile '/tpch_df/l_67' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l68 tablespace
*sql
{
drop tablespace ts_l68 including contents;
create tablespace ts_l68
datafile '/tpch_df/l_68' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l69 tablespace
*sql
{
drop tablespace ts_l69 including contents;
create tablespace ts_l69
datafile '/tpch_df/l_69' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l70 tablespace
*sql
{
drop tablespace ts_l70 including contents;
create tablespace ts_l70
datafile '/tpch_df/l_70' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l71 tablespace
*sql
{
drop tablespace ts_l71 including contents;
create tablespace ts_l71
datafile '/tpch_df/l_71' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l72 tablespace
*sql
{
drop tablespace ts_l72 including contents;
create tablespace ts_l72
datafile '/tpch_df/l_72' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l73 tablespace
```

```
*sql
{
drop tablespace ts_l73 including contents;
create tablespace ts_l73
datafile '/tpch_df/l_73' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l74 tablespace
*sql
{
drop tablespace ts_l74 including contents;
create tablespace ts_l74
datafile '/tpch_df/l_74' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l75 tablespace
*sql
{
drop tablespace ts_l75 including contents;
create tablespace ts_l75
datafile '/tpch_df/l_75' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l76 tablespace
*sql
{
drop tablespace ts_l76 including contents;
create tablespace ts_l76
datafile '/tpch_df/l_76' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l77 tablespace
*sql
{
drop tablespace ts_l77 including contents;
create tablespace ts_l77
datafile '/tpch_df/l_77' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l78 tablespace
*sql
{
drop tablespace ts_l78 including contents;
create tablespace ts_l78
datafile '/tpch_df/l_78' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l79 tablespace
*sql
{
drop tablespace ts_l79 including contents;
create tablespace ts_l79
datafile '/tpch_df/l_79' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l80 tablespace
*sql
{
drop tablespace ts_l80 including contents;
create tablespace ts_l80
datafile '/tpch_df/l_80' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l81 tablespace
*sql
{
drop tablespace ts_l81 including contents;
create tablespace ts_l81
datafile '/tpch_df/l_81' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l82 tablespace
*sql
{
drop tablespace ts_l82 including contents;
create tablespace ts_l82
datafile '/tpch_df/l_82' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l83 tablespace
*sql
{
drop tablespace ts_l83 including contents;
create tablespace ts_l83
datafile '/tpch_df/l_83' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_l84 tablespace
*sql
{
drop tablespace ts_l84 including contents;
create tablespace ts_l84
datafile '/tpch_df/l_84' size 14405m reuse
extent management local
autoallocate
;
}
# creating tpch's ts_okey tablespace
*sql
{
drop tablespace ts_okey including contents;
create tablespace ts_okey
datafile '/tpch_df/okey_1' size 26540m reuse
extent management local
autoallocate
 nologging
;
}
# creating tpch's ts_lokey tablespace
*sql
{
drop tablespace ts_lokey including contents;
create tablespace ts_lokey
datafile '/tpch_df/lokey_1' size 30900m reuse
extent management local
autoallocate
 nologging
;
}
# creating tpch's ts_ckey tablespace
*sql
{
```

```
drop tablespace ts_ckey including contents;
create tablespace ts_ckey
datafile '/tpch_df/ckey_1' size 22510m reuse
extent management local
autoallocate
 nologging
;
}
# creating tpch's ts_temp tablespace
*sql
{
drop tablespace ts_temp including contents;
create temporary tablespace ts_temp
tempfile '/tpch_df/tmp_1' size 29550m reuse
extent management local
uniform size 10m
;
}
*wait
# adding tpch's ts_s datafiles
# adding tpch's ts_c datafiles
# adding tpch's ts_undo1 datafiles
*sql
{
alter tablespace ts_undo1
  add datafile '/tpch_df/undo_2' size 30570m
reuse;
}
*sql
{
alter tablespace ts_undo1
  add datafile '/tpch_df/undo_3' size 30570m
reuse;
}
*sql
{
alter tablespace ts_undo1
  add datafile '/tpch_df/undo_4' size 30570m
reuse;
}
# adding tpch's ts_ps datafiles
*sql
{
alter tablespace ts_ps
  add datafile '/tpch_df/ps_2' size 29010m
reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile '/tpch_df/ps_3' size 29010m
reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile '/tpch_df/ps_4' size 29010m
reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile '/tpch_df/ps_5' size 29010m
reuse;
}
# adding tpch's ts_p datafiles
# adding tpch's ts_okey datafiles
*sql
{
alter tablespace ts_okey
  add datafile '/tpch_df/okey_2' size 26540m
reuse;
}
# adding tpch's ts_lokey datafiles
*sql
{
alter tablespace ts_lokey
  add datafile '/tpch_df/lokey_2' size 30900m
reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/tpch_df/lokey_3' size 30900m
reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/tpch_df/lokey_4' size 30900m
reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/tpch_df/lokey_5' size 30900m
reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/tpch_df/lokey_6' size 30900m
reuse;
}
# adding tpch's ts_ckey datafiles
# adding tpch's ts_temp datafiles
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_2' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_3' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_4' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_5' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_6' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_7' size 29560m
reuse;
}
*sql
{
```

```
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_8' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_9' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_10' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_11' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_12' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_13' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_14' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_15' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_16' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_17' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_18' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_19' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_20' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_21' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_22' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_23' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_24' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_25' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_26' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_27' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_28' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_29' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_30' size 29560m
reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_31' size 29560m
reuse;
}
*sql
{
```

```
alter tablespace ts_temp
  add tempfile '/tpch_df/tmp_32' size 29560m
reuse;
}
*wait
*wait
*bgoff
%e-sctso
%b-dapop
*bgon=1
###########################################
##################################
# Schema Creation Phase - User and Tables
#  AND  Database Population Phase
*sql
{
shutdown;
startup pfile =
/export/home/oracle/tpch/admin/init_build.o
ra
}
*wait
# creating tpch user
*sql
{
drop user tpch cascade;
grant DBA
  to tpch identified by tpch;
}
*wait
*sql
{
connect tpch/tpch;
}
*wait
# altering tpch's temp and default tablespace
*sql
{
alter user tpch temporary tablespace ts_temp;
alter user tpch default tablespace ts_def;
}
*wait
*sql
{
connect tpch/tpch
@?/rdbms/admin/utlxplan.sql;
}
*wait
# External Tables Definition Phase
*sql
{
connect tpch/tpch;
drop directory data_dir;
create directory data_dir as '/flat';
}
*wait
*sql
{
connect tpch/tpch;
drop table lineitem_et;
create table lineitem_et (
    l_orderkey          number ,
    l_partkey           number ,
    l_suppkey           number ,
    l_linenumber         number ,
    l_quantity          number ,
    l_extendedprice      number ,
    l_discount          number ,
    l_tax              number ,
    l_returnflag         char(1) ,
    l_linestatus         char(1) ,
    l_shipdate          date ,
    l_commitdate         date ,
    l_receiptdate        date ,
    l_shipinstruct        char(25) ,
    l_shipmode          char(10) ,
    l_comment           varchar(44)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
       records delimited by newline
       badfile 'lineitem.bad'
       logfile 'lineitem.log'
       nodiscardfile
       fields terminated by '|'
       missing field values are null
)
location (
'lineitem.tbl.1','lineitem.tbl.2','lineitem
.tbl.3',
'lineitem.tbl.4','lineitem.tbl.5','lineitem
.tbl.6',
'lineitem.tbl.7','lineitem.tbl.8','lineitem
.tbl.9',
'lineitem.tbl.10','lineitem.tbl.11','lineit
em.tbl.12',
'lineitem.tbl.13','lineitem.tbl.14','lineit
em.tbl.15',
'lineitem.tbl.16','lineitem.tbl.17','lineit
em.tbl.18',
'lineitem.tbl.19','lineitem.tbl.20','lineit
em.tbl.21',
'lineitem.tbl.22','lineitem.tbl.23','lineit
em.tbl.24',
'lineitem.tbl.25','lineitem.tbl.26','lineit
em.tbl.27',
'lineitem.tbl.28','lineitem.tbl.29','lineit
em.tbl.30',
'lineitem.tbl.31','lineitem.tbl.32','lineit
em.tbl.33',
'lineitem.tbl.34','lineitem.tbl.35','lineit
em.tbl.36',
'lineitem.tbl.37','lineitem.tbl.38','lineit
em.tbl.39',
'lineitem.tbl.40','lineitem.tbl.41','lineit
em.tbl.42',
'lineitem.tbl.43','lineitem.tbl.44','lineit
em.tbl.45',
'lineitem.tbl.46','lineitem.tbl.47','lineit
em.tbl.48',
'lineitem.tbl.49','lineitem.tbl.50','lineit
em.tbl.51',
'lineitem.tbl.52','lineitem.tbl.53','lineit
em.tbl.54',
'lineitem.tbl.55','lineitem.tbl.56','lineit
em.tbl.57',
'lineitem.tbl.58','lineitem.tbl.59','lineit
em.tbl.60',
'lineitem.tbl.61','lineitem.tbl.62','lineit
em.tbl.63',
'lineitem.tbl.64','lineitem.tbl.65','lineit
em.tbl.66',
'lineitem.tbl.67','lineitem.tbl.68','lineit
em.tbl.69',
'lineitem.tbl.70','lineitem.tbl.71','lineit
em.tbl.72',
'lineitem.tbl.73','lineitem.tbl.74','lineit
em.tbl.75',
'lineitem.tbl.76','lineitem.tbl.77','lineit
em.tbl.78',
'lineitem.tbl.79','lineitem.tbl.80','lineit
em.tbl.81',
'lineitem.tbl.82','lineitem.tbl.83','lineit
em.tbl.84'

))reject limit unlimited;
alter table lineitem_et parallel;
}
*sql
{
connect tpch/tpch;
drop table orders_et;
create table orders_et (
   o_orderkey         number ,
   o_custkey          number ,
   o_orderstatus       char(1) ,
   o_totalprice        number ,
   o_orderdate        date ,
   o_orderpriority     char(15) ,
   o_clerk            char(15) ,
   o_shippriority      number ,
   o_comment          varchar(79)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
      records delimited by newline
      badfile 'orders.bad'
      logfile 'orders.log'
      nodiscardfile
      fields terminated by '|'
      missing field values are null
)
location (
'orders.tbl.1','orders.tbl.2','orders.tbl.3
',
'orders.tbl.4','orders.tbl.5','orders.tbl.6
',
'orders.tbl.7','orders.tbl.8','orders.tbl.9
',
'orders.tbl.10','orders.tbl.11','orders.tbl
.12',
'orders.tbl.13','orders.tbl.14','orders.tbl
.15',
'orders.tbl.16','orders.tbl.17','orders.tbl
.18',
'orders.tbl.19','orders.tbl.20','orders.tbl
.21',
'orders.tbl.22','orders.tbl.23','orders.tbl
.24',
'orders.tbl.25','orders.tbl.26','orders.tbl
.27',
'orders.tbl.28','orders.tbl.29','orders.tbl
.30',
'orders.tbl.31','orders.tbl.32','orders.tbl
.33',
'orders.tbl.34','orders.tbl.35','orders.tbl
.36',
'orders.tbl.37','orders.tbl.38','orders.tbl
.39',
'orders.tbl.40','orders.tbl.41','orders.tbl
.42',
'orders.tbl.43','orders.tbl.44','orders.tbl
.45',
'orders.tbl.46','orders.tbl.47','orders.tbl
.48',
'orders.tbl.49','orders.tbl.50','orders.tbl
.51',
'orders.tbl.52','orders.tbl.53','orders.tbl
.54',
'orders.tbl.55','orders.tbl.56','orders.tbl
.57',
'orders.tbl.58','orders.tbl.59','orders.tbl
.60',
'orders.tbl.61','orders.tbl.62','orders.tbl
.63',
'orders.tbl.64','orders.tbl.65','orders.tbl
.66',
'orders.tbl.67','orders.tbl.68','orders.tbl
.69',
'orders.tbl.70','orders.tbl.71','orders.tbl
.72',
'orders.tbl.73','orders.tbl.74','orders.tbl
.75',
'orders.tbl.76','orders.tbl.77','orders.tbl
.78',
'orders.tbl.79','orders.tbl.80','orders.tbl
.81',
'orders.tbl.82','orders.tbl.83','orders.tbl
.84'

))reject limit unlimited;
alter table orders_et parallel;
}
*sql
{
connect tpch/tpch;
drop table part_et;
create table part_et (
   p_partkey          number ,
   p_name            varchar(55) ,
   p_mfgr            char(25) ,
   p_brand            char(10) ,
   p_type            varchar(25) ,
   p_size            number ,
   p_container        char(10) ,
   p_retailprice       number ,
   p_comment          varchar(23)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
      records delimited by newline
      badfile 'part.bad'
      logfile 'part.log'
      nodiscardfile
      fields terminated by '|'
      missing field values are null
)
location (
'part.tbl.1','part.tbl.2','part.tbl.3',
'part.tbl.4','part.tbl.5','part.tbl.6',
'part.tbl.7','part.tbl.8','part.tbl.9',
'part.tbl.10','part.tbl.11','part.tbl.12',
'part.tbl.13','part.tbl.14','part.tbl.15',
'part.tbl.16','part.tbl.17','part.tbl.18',
'part.tbl.19','part.tbl.20','part.tbl.21',
'part.tbl.22','part.tbl.23','part.tbl.24',
'part.tbl.25','part.tbl.26','part.tbl.27',
'part.tbl.28','part.tbl.29','part.tbl.30',
'part.tbl.31','part.tbl.32'
))reject limit unlimited;
alter table part_et parallel;
}
*sql
{
connect tpch/tpch;
drop table partsupp_et;
create table partsupp_et (
   ps_partkey          number ,
   ps_suppkey         number ,
   ps_availqty        number ,
   ps_supplycost      number ,
   ps_comment         varchar(199)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
      records delimited by newline
      badfile 'partsupp.bad'
      logfile 'partsupp.log'
      nodiscardfile
      fields terminated by '|'
      missing field values are null
)
location (
'partsupp.tbl.1','partsupp.tbl.2','partsupp
.tbl.3',
'partsupp.tbl.4','partsupp.tbl.5','partsupp
.tbl.6',
'partsupp.tbl.7','partsupp.tbl.8','partsupp
.tbl.9',
'partsupp.tbl.10','partsupp.tbl.11','partsu
pp.tbl.12',
'partsupp.tbl.13','partsupp.tbl.14','partsu
pp.tbl.15',
'partsupp.tbl.16','partsupp.tbl.17','partsu
pp.tbl.18',
'partsupp.tbl.19','partsupp.tbl.20','partsu
pp.tbl.21',
'partsupp.tbl.22','partsupp.tbl.23','partsu
pp.tbl.24',
'partsupp.tbl.25','partsupp.tbl.26','partsu
pp.tbl.27',
'partsupp.tbl.28','partsupp.tbl.29','partsu
pp.tbl.30',
'partsupp.tbl.31','partsupp.tbl.32'
))reject limit unlimited;
alter table partsupp_et parallel;
}
*sql
{
connect tpch/tpch;
drop table supplier_et;
create table supplier_et (
   s_suppkey          number ,
   s_name            char(25) ,
   s_address          varchar(40) ,
   s_nationkey        number ,
   s_phone           char(15) ,
   s_acctbal          number ,
   s_comment          varchar(101)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
      records delimited by newline
      badfile 'supplier.bad'
      logfile 'supplier.log'
      nodiscardfile
      fields terminated by '|'
      missing field values are null
)
location (
'supplier.tbl'
))reject limit unlimited;
alter table supplier_et parallel;
}
```

```
*sql
{
connect tpch/tpch;
drop table customer_et;
create table customer_et (
   c_custkey          number ,
   c_name            varchar(25) ,
   c_address          varchar(40) ,
   c_nationkey        number ,
   c_phone           char(15) ,
   c_acctbal          number ,
   c_mktsegment       char(10) ,
   c_comment          varchar(117)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
      records delimited by newline
      badfile 'customer.bad'
      logfile 'customer.log'
      nodiscardfile
      fields terminated by '|'
      missing field values are null
)
location (
'customer.tbl.1','customer.tbl.2','customer
.tbl.3',
'customer.tbl.4','customer.tbl.5','customer
.tbl.6',
'customer.tbl.7','customer.tbl.8'
))reject limit unlimited;
alter table customer_et parallel;
}
*sql
{
connect tpch/tpch;
drop table nation_et;
create table nation_et (
   n_nationkey        number ,
   n_name            char(25) ,
   n_regionkey        number ,
   n_comment          varchar(152)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
      records delimited by newline
      badfile 'nation.bad'
      logfile 'nation.log'
      nodiscardfile
      fields terminated by '|'
      missing field values are null
)
location (
'nation.tbl'
))reject limit unlimited;
alter table nation_et parallel;
}
*sql
{
connect tpch/tpch;
drop table region_et;
create table region_et (
   r_regionkey        number ,
   r_name            char(25) ,
   r_comment          varchar(152)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
       records delimited by newline
       badfile 'region.bad'
       logfile 'region.log'
       nodiscardfile
       fields terminated by '|'
       missing field values are null
)
location (
'region.tbl'
))reject limit unlimited;
alter table region_et parallel;
}
###########################################
##################################
# Schema Creation Phase - User and Tables ONLY
(no datafiles)
*wait
*sql
{
connect tpch/tpch;
}
*wait
*sql
{
connect tpch/tpch
@?/rdbms/admin/utlxplan.sql;
}
*wait
*sql
{
connect tpch/tpch;
set timing on
set echo on
!date

rem drop table lineitem;
create table lineitem(
   l_shipdate          ,
   l_orderkey         NOT NULL,
   l_discount         NOT NULL,
   l_extendedprice     NOT NULL,
   l_suppkey          NOT NULL,
   l_quantity         NOT NULL,
   l_returnflag        ,
   l_partkey          NOT NULL,
   l_linestatus        ,
   l_tax             NOT NULL,
   l_commitdate        ,
   l_receiptdate       ,
   l_shipmode          ,
   l_linenumber        NOT NULL,
   l_shipinstruct       ,
   l_comment
)
pctfree 1
pctused 99
initrans 10
storage (initial 800m freelist groups 4
freelists 84)
parallel
nologging
partition by range (l_shipdate)
subpartition by hash(l_partkey)
subpartitions 16
(
partition item1 values less than
(to_date('1992-01-01','YYYY-MM-DD'))
tablespace ts_l1
```

```
,
partition item2 values less than
(to_date('1992-02-01','YYYY-MM-DD'))
tablespace ts_l2
,
partition item3 values less than
(to_date('1992-03-01','YYYY-MM-DD'))
tablespace ts_l3
,
partition item4 values less than
(to_date('1992-04-01','YYYY-MM-DD'))
tablespace ts_l4
,
partition item5 values less than
(to_date('1992-05-01','YYYY-MM-DD'))
tablespace ts_l5
,
partition item6 values less than
(to_date('1992-06-01','YYYY-MM-DD'))
tablespace ts_l6
,
partition item7 values less than
(to_date('1992-07-01','YYYY-MM-DD'))
tablespace ts_l7
,
partition item8 values less than
(to_date('1992-08-01','YYYY-MM-DD'))
tablespace ts_l8
,
partition item9 values less than
(to_date('1992-09-01','YYYY-MM-DD'))
tablespace ts_l9
,
partition item10 values less than
(to_date('1992-10-01','YYYY-MM-DD'))
tablespace ts_l10
,
partition item11 values less than
(to_date('1992-11-01','YYYY-MM-DD'))
tablespace ts_l11
,
partition item12 values less than
(to_date('1992-12-01','YYYY-MM-DD'))
tablespace ts_l12
,
partition item13 values less than
(to_date('1993-01-01','YYYY-MM-DD'))
tablespace ts_l13
,
partition item14 values less than
(to_date('1993-02-01','YYYY-MM-DD'))
tablespace ts_l14
,
partition item15 values less than
(to_date('1993-03-01','YYYY-MM-DD'))
tablespace ts_l15
,
partition item16 values less than
(to_date('1993-04-01','YYYY-MM-DD'))
tablespace ts_l16
,
partition item17 values less than
(to_date('1993-05-01','YYYY-MM-DD'))
tablespace ts_l17
,
partition item18 values less than
(to_date('1993-06-01','YYYY-MM-DD'))
tablespace ts_l18
,
partition item19 values less than
(to_date('1993-07-01','YYYY-MM-DD'))
tablespace ts_l19
```

```
,
partition item20 values less than
(to_date('1993-08-01','YYYY-MM-DD'))
tablespace ts_l20
,
partition item21 values less than
(to_date('1993-09-01','YYYY-MM-DD'))
tablespace ts_l21
,
partition item22 values less than
(to_date('1993-10-01','YYYY-MM-DD'))
tablespace ts_l22
,
partition item23 values less than
(to_date('1993-11-01','YYYY-MM-DD'))
tablespace ts_l23
,
partition item24 values less than
(to_date('1993-12-01','YYYY-MM-DD'))
tablespace ts_l24
,
partition item25 values less than
(to_date('1994-01-01','YYYY-MM-DD'))
tablespace ts_l25
,
partition item26 values less than
(to_date('1994-02-01','YYYY-MM-DD'))
tablespace ts_l26
,
partition item27 values less than
(to_date('1994-03-01','YYYY-MM-DD'))
tablespace ts_l27
,
partition item28 values less than
(to_date('1994-04-01','YYYY-MM-DD'))
tablespace ts_l28
,
partition item29 values less than
(to_date('1994-05-01','YYYY-MM-DD'))
tablespace ts_l29
,
partition item30 values less than
(to_date('1994-06-01','YYYY-MM-DD'))
tablespace ts_l30
,
partition item31 values less than
(to_date('1994-07-01','YYYY-MM-DD'))
tablespace ts_l31
,
partition item32 values less than
(to_date('1994-08-01','YYYY-MM-DD'))
tablespace ts_l32
,
partition item33 values less than
(to_date('1994-09-01','YYYY-MM-DD'))
tablespace ts_l33
,
partition item34 values less than
(to_date('1994-10-01','YYYY-MM-DD'))
tablespace ts_l34
,
partition item35 values less than
(to_date('1994-11-01','YYYY-MM-DD'))
tablespace ts_l35
,
partition item36 values less than
(to_date('1994-12-01','YYYY-MM-DD'))
tablespace ts_l36
,
partition item37 values less than
(to_date('1995-01-01','YYYY-MM-DD'))
tablespace ts_l37
,
partition item38 values less than
(to_date('1995-02-01','YYYY-MM-DD'))
tablespace ts_l38
,
partition item39 values less than
(to_date('1995-03-01','YYYY-MM-DD'))
tablespace ts_l39
,
partition item40 values less than
(to_date('1995-04-01','YYYY-MM-DD'))
tablespace ts_l40
,
partition item41 values less than
(to_date('1995-05-01','YYYY-MM-DD'))
tablespace ts_l41
,
partition item42 values less than
(to_date('1995-06-01','YYYY-MM-DD'))
tablespace ts_l42
,
partition item43 values less than
(to_date('1995-07-01','YYYY-MM-DD'))
tablespace ts_l43
,
partition item44 values less than
(to_date('1995-08-01','YYYY-MM-DD'))
tablespace ts_l44
,
partition item45 values less than
(to_date('1995-09-01','YYYY-MM-DD'))
tablespace ts_l45
,
partition item46 values less than
(to_date('1995-10-01','YYYY-MM-DD'))
tablespace ts_l46
,
partition item47 values less than
(to_date('1995-11-01','YYYY-MM-DD'))
tablespace ts_l47
,
partition item48 values less than
(to_date('1995-12-01','YYYY-MM-DD'))
tablespace ts_l48
,
partition item49 values less than
(to_date('1996-01-01','YYYY-MM-DD'))
tablespace ts_l49
,
partition item50 values less than
(to_date('1996-02-01','YYYY-MM-DD'))
tablespace ts_l50
,
partition item51 values less than
(to_date('1996-03-01','YYYY-MM-DD'))
tablespace ts_l51
,
partition item52 values less than
(to_date('1996-04-01','YYYY-MM-DD'))
tablespace ts_l52
,
partition item53 values less than
(to_date('1996-05-01','YYYY-MM-DD'))
tablespace ts_l53
,
partition item54 values less than
(to_date('1996-06-01','YYYY-MM-DD'))
tablespace ts_l54
,
partition item55 values less than
(to_date('1996-07-01','YYYY-MM-DD'))
tablespace ts_l55
```

```
,
partition item56 values less than
(to_date('1996-08-01','YYYY-MM-DD'))
tablespace ts_l56
,
partition item57 values less than
(to_date('1996-09-01','YYYY-MM-DD'))
tablespace ts_l57
,
partition item58 values less than
(to_date('1996-10-01','YYYY-MM-DD'))
tablespace ts_l58
,
partition item59 values less than
(to_date('1996-11-01','YYYY-MM-DD'))
tablespace ts_l59
,
partition item60 values less than
(to_date('1996-12-01','YYYY-MM-DD'))
tablespace ts_l60
,
partition item61 values less than
(to_date('1997-01-01','YYYY-MM-DD'))
tablespace ts_l61
,
partition item62 values less than
(to_date('1997-02-01','YYYY-MM-DD'))
tablespace ts_l62
,
partition item63 values less than
(to_date('1997-03-01','YYYY-MM-DD'))
tablespace ts_l63
,
partition item64 values less than
(to_date('1997-04-01','YYYY-MM-DD'))
tablespace ts_l64
,
partition item65 values less than
(to_date('1997-05-01','YYYY-MM-DD'))
tablespace ts_l65
,
partition item66 values less than
(to_date('1997-06-01','YYYY-MM-DD'))
tablespace ts_l66
,
partition item67 values less than
(to_date('1997-07-01','YYYY-MM-DD'))
tablespace ts_l67
,
partition item68 values less than
(to_date('1997-08-01','YYYY-MM-DD'))
tablespace ts_l68
,
partition item69 values less than
(to_date('1997-09-01','YYYY-MM-DD'))
tablespace ts_l69
,
partition item70 values less than
(to_date('1997-10-01','YYYY-MM-DD'))
tablespace ts_l70
,
partition item71 values less than
(to_date('1997-11-01','YYYY-MM-DD'))
tablespace ts_l71
,
partition item72 values less than
(to_date('1997-12-01','YYYY-MM-DD'))
tablespace ts_l72
,
partition item73 values less than
(to_date('1998-01-01','YYYY-MM-DD'))
tablespace ts_l73
```

```
,
partition item74 values less than
(to_date('1998-02-01','YYYY-MM-DD'))
tablespace ts_l74
,
partition item75 values less than
(to_date('1998-03-01','YYYY-MM-DD'))
tablespace ts_l75
,
partition item76 values less than
(to_date('1998-04-01','YYYY-MM-DD'))
tablespace ts_l76
,
partition item77 values less than
(to_date('1998-05-01','YYYY-MM-DD'))
tablespace ts_l77
,
partition item78 values less than
(to_date('1998-06-01','YYYY-MM-DD'))
tablespace ts_l78
,
partition item79 values less than
(to_date('1998-07-01','YYYY-MM-DD'))
tablespace ts_l79
,
partition item80 values less than
(to_date('1998-08-01','YYYY-MM-DD'))
tablespace ts_l80
,
partition item81 values less than
(to_date('1998-09-01','YYYY-MM-DD'))
tablespace ts_l81
,
partition item82 values less than
(to_date('1998-10-01','YYYY-MM-DD'))
tablespace ts_l82
,
partition item83 values less than
(to_date('1998-11-01','YYYY-MM-DD'))
tablespace ts_l83
,
partition item84 values less than (MAXVALUE)
tablespace ts_l84
)
as select
   l_shipdate         ,
   l_orderkey         ,
   l_discount         ,
   l_extendedprice     ,
   l_suppkey          ,
   l_quantity         ,
   l_returnflag        ,
   l_partkey          ,
   l_linestatus        ,
   l_tax             ,
   l_commitdate        ,
   l_receiptdate       ,
   l_shipmode         ,
   l_linenumber        ,
   l_shipinstruct      ,
   l_comment
from lineitem_et;
rem drop table lineitem_et;
}
*wait
*sql
{
connect tpch/tpch;
set timing on
set echo on
!date
```

```
rem drop table orders;
create table orders(
   o_orderdate         ,
   o_orderkey         NOT NULL,
   o_custkey          NOT NULL,
   o_orderpriority     ,
   o_shippriority      ,
   o_clerk            ,
   o_orderstatus       ,
   o_totalprice        ,
   o_comment
)
pctfree 1
pctused 99
initrans 10
storage (initial 150m freelist groups 4
freelists 84)
parallel
nologging
partition by range (o_orderdate)
subpartition by hash(o_custkey)
subpartitions 16
(
partition ord1 values less than
(to_date('1992-01-01','YYYY-MM-DD'))
tablespace ts_o1
,
partition ord2 values less than
(to_date('1992-02-01','YYYY-MM-DD'))
tablespace ts_o2
,
partition ord3 values less than
(to_date('1992-03-01','YYYY-MM-DD'))
tablespace ts_o3
,
partition ord4 values less than
(to_date('1992-04-01','YYYY-MM-DD'))
tablespace ts_o4
,
partition ord5 values less than
(to_date('1992-05-01','YYYY-MM-DD'))
tablespace ts_o5
,
partition ord6 values less than
(to_date('1992-06-01','YYYY-MM-DD'))
tablespace ts_o6
,
partition ord7 values less than
(to_date('1992-07-01','YYYY-MM-DD'))
tablespace ts_o7
,
partition ord8 values less than
(to_date('1992-08-01','YYYY-MM-DD'))
tablespace ts_o8
,
partition ord9 values less than
(to_date('1992-09-01','YYYY-MM-DD'))
tablespace ts_o9
,
partition ord10 values less than
(to_date('1992-10-01','YYYY-MM-DD'))
tablespace ts_o10
,
partition ord11 values less than
(to_date('1992-11-01','YYYY-MM-DD'))
tablespace ts_o11
,
partition ord12 values less than
(to_date('1992-12-01','YYYY-MM-DD'))
tablespace ts_o12
,
partition ord13 values less than
(to_date('1993-01-01','YYYY-MM-DD'))
tablespace ts_o13
,
partition ord14 values less than
(to_date('1993-02-01','YYYY-MM-DD'))
tablespace ts_o14
,
partition ord15 values less than
(to_date('1993-03-01','YYYY-MM-DD'))
tablespace ts_o15
,
partition ord16 values less than
(to_date('1993-04-01','YYYY-MM-DD'))
tablespace ts_o16
,
partition ord17 values less than
(to_date('1993-05-01','YYYY-MM-DD'))
tablespace ts_o17
,
partition ord18 values less than
(to_date('1993-06-01','YYYY-MM-DD'))
tablespace ts_o18
,
partition ord19 values less than
(to_date('1993-07-01','YYYY-MM-DD'))
tablespace ts_o19
,
partition ord20 values less than
(to_date('1993-08-01','YYYY-MM-DD'))
tablespace ts_o20
,
partition ord21 values less than
(to_date('1993-09-01','YYYY-MM-DD'))
tablespace ts_o21
,
partition ord22 values less than
(to_date('1993-10-01','YYYY-MM-DD'))
tablespace ts_o22
,
partition ord23 values less than
(to_date('1993-11-01','YYYY-MM-DD'))
tablespace ts_o23
,
partition ord24 values less than
(to_date('1993-12-01','YYYY-MM-DD'))
tablespace ts_o24
,
partition ord25 values less than
(to_date('1994-01-01','YYYY-MM-DD'))
tablespace ts_o25
,
partition ord26 values less than
(to_date('1994-02-01','YYYY-MM-DD'))
tablespace ts_o26
,
partition ord27 values less than
(to_date('1994-03-01','YYYY-MM-DD'))
tablespace ts_o27
,
partition ord28 values less than
(to_date('1994-04-01','YYYY-MM-DD'))
tablespace ts_o28
,
partition ord29 values less than
(to_date('1994-05-01','YYYY-MM-DD'))
tablespace ts_o29
,
partition ord30 values less than
(to_date('1994-06-01','YYYY-MM-DD'))
tablespace ts_o30
,
partition ord31 values less than
(to_date('1994-07-01','YYYY-MM-DD'))
tablespace ts_o31
,
partition ord32 values less than
(to_date('1994-08-01','YYYY-MM-DD'))
tablespace ts_o32
,
partition ord33 values less than
(to_date('1994-09-01','YYYY-MM-DD'))
tablespace ts_o33
,
partition ord34 values less than
(to_date('1994-10-01','YYYY-MM-DD'))
tablespace ts_o34
,
partition ord35 values less than
(to_date('1994-11-01','YYYY-MM-DD'))
tablespace ts_o35
,
partition ord36 values less than
(to_date('1994-12-01','YYYY-MM-DD'))
tablespace ts_o36
,
partition ord37 values less than
(to_date('1995-01-01','YYYY-MM-DD'))
tablespace ts_o37
,
partition ord38 values less than
(to_date('1995-02-01','YYYY-MM-DD'))
tablespace ts_o38
,
partition ord39 values less than
(to_date('1995-03-01','YYYY-MM-DD'))
tablespace ts_o39
,
partition ord40 values less than
(to_date('1995-04-01','YYYY-MM-DD'))
tablespace ts_o40
,
partition ord41 values less than
(to_date('1995-05-01','YYYY-MM-DD'))
tablespace ts_o41
,
partition ord42 values less than
(to_date('1995-06-01','YYYY-MM-DD'))
tablespace ts_o42
,
partition ord43 values less than
(to_date('1995-07-01','YYYY-MM-DD'))
tablespace ts_o43
,
partition ord44 values less than
(to_date('1995-08-01','YYYY-MM-DD'))
tablespace ts_o44
,
partition ord45 values less than
(to_date('1995-09-01','YYYY-MM-DD'))
tablespace ts_o45
,
partition ord46 values less than
(to_date('1995-10-01','YYYY-MM-DD'))
tablespace ts_o46
,
partition ord47 values less than
(to_date('1995-11-01','YYYY-MM-DD'))
tablespace ts_o47
,
partition ord48 values less than
(to_date('1995-12-01','YYYY-MM-DD'))
tablespace ts_o48
,
partition ord49 values less than
(to_date('1996-01-01','YYYY-MM-DD'))
tablespace ts_o49
,
partition ord50 values less than
(to_date('1996-02-01','YYYY-MM-DD'))
tablespace ts_o50
,
partition ord51 values less than
(to_date('1996-03-01','YYYY-MM-DD'))
tablespace ts_o51
,
partition ord52 values less than
(to_date('1996-04-01','YYYY-MM-DD'))
tablespace ts_o52
,
partition ord53 values less than
(to_date('1996-05-01','YYYY-MM-DD'))
tablespace ts_o53
,
partition ord54 values less than
(to_date('1996-06-01','YYYY-MM-DD'))
tablespace ts_o54
,
partition ord55 values less than
(to_date('1996-07-01','YYYY-MM-DD'))
tablespace ts_o55
,
partition ord56 values less than
(to_date('1996-08-01','YYYY-MM-DD'))
tablespace ts_o56
,
partition ord57 values less than
(to_date('1996-09-01','YYYY-MM-DD'))
tablespace ts_o57
,
partition ord58 values less than
(to_date('1996-10-01','YYYY-MM-DD'))
tablespace ts_o58
,
partition ord59 values less than
(to_date('1996-11-01','YYYY-MM-DD'))
tablespace ts_o59
,
partition ord60 values less than
(to_date('1996-12-01','YYYY-MM-DD'))
tablespace ts_o60
,
partition ord61 values less than
(to_date('1997-01-01','YYYY-MM-DD'))
tablespace ts_o61
,
partition ord62 values less than
(to_date('1997-02-01','YYYY-MM-DD'))
tablespace ts_o62
,
partition ord63 values less than
(to_date('1997-03-01','YYYY-MM-DD'))
tablespace ts_o63
,
partition ord64 values less than
(to_date('1997-04-01','YYYY-MM-DD'))
tablespace ts_o64
,
partition ord65 values less than
(to_date('1997-05-01','YYYY-MM-DD'))
tablespace ts_o65
,
partition ord66 values less than
(to_date('1997-06-01','YYYY-MM-DD'))
tablespace ts_o66
,
partition ord67 values less than
(to_date('1997-07-01','YYYY-MM-DD'))
tablespace ts_o67
,
partition ord68 values less than
(to_date('1997-08-01','YYYY-MM-DD'))
tablespace ts_o68
,
partition ord69 values less than
(to_date('1997-09-01','YYYY-MM-DD'))
tablespace ts_o69
,
partition ord70 values less than
(to_date('1997-10-01','YYYY-MM-DD'))
tablespace ts_o70
,
partition ord71 values less than
(to_date('1997-11-01','YYYY-MM-DD'))
tablespace ts_o71
,
partition ord72 values less than
(to_date('1997-12-01','YYYY-MM-DD'))
tablespace ts_o72
,
partition ord73 values less than
(to_date('1998-01-01','YYYY-MM-DD'))
tablespace ts_o73
,
partition ord74 values less than
(to_date('1998-02-01','YYYY-MM-DD'))
tablespace ts_o74
,
partition ord75 values less than
(to_date('1998-03-01','YYYY-MM-DD'))
tablespace ts_o75
,
partition ord76 values less than
(to_date('1998-04-01','YYYY-MM-DD'))
tablespace ts_o76
,
partition ord77 values less than
(to_date('1998-05-01','YYYY-MM-DD'))
tablespace ts_o77
,
partition ord78 values less than
(to_date('1998-06-01','YYYY-MM-DD'))
tablespace ts_o78
,
partition ord79 values less than
(to_date('1998-07-01','YYYY-MM-DD'))
tablespace ts_o79
,
partition ord80 values less than
(to_date('1998-08-01','YYYY-MM-DD'))
tablespace ts_o80
,
partition ord81 values less than
(to_date('1998-09-01','YYYY-MM-DD'))
tablespace ts_o81
,
partition ord82 values less than
(to_date('1998-10-01','YYYY-MM-DD'))
tablespace ts_o82
,
partition ord83 values less than
(to_date('1998-11-01','YYYY-MM-DD'))
tablespace ts_o83
,
partition ord84 values less than (MAXVALUE)
tablespace ts_o84
)
as select
   o_orderdate        ,
   o_orderkey         ,
   o_custkey          ,
   o_orderpriority     ,
   o_shippriority      ,
   o_clerk            ,
   o_orderstatus       ,
   o_totalprice        ,
   o_comment
from orders_et;
rem drop table orders_et;
}
*wait
*sql
{
connect tpch/tpch;
set timing on
set echo on
!date
-- alter session force parallel dml parallel;
-- alter session set
isolation_level=serializable;

rem drop table partsupp;
create table partsupp(
   ps_partkey         NOT NULL,
   ps_suppkey         NOT NULL,
   ps_supplycost       NOT NULL,
   ps_availqty         ,
   ps_comment          ,
   constraint pk_partkey_suppkey_1 primary
key(ps_partkey,ps_suppkey)
)
organization index
pctthreshold 50
tablespace ts_ps
compress
storage (initial 1000m)
parallel
nologging
partition by hash (ps_partkey)
partitions 128
as select
   ps_partkey         ,
   ps_suppkey         ,
   ps_supplycost       ,
   ps_availqty         ,
   ps_comment
from partsupp_et;
rem drop table partsupp_et;
}
*wait
*sql
{
connect tpch/tpch;
set timing on
set echo on
!date

rem drop table part;
create table part(
   p_partkey          NOT NULL,
   p_type            ,
   p_size            ,
   p_brand           ,
   p_name            ,
   p_container        ,
   p_mfgr            ,
   p_retailprice       ,
   p_comment
)
pctfree 0
```

```
pctused 99
tablespace ts_p
parallel
storage (initial 200m freelists 84)
nologging
partition by hash (p_partkey)
partitions 128
as select
   p_partkey          ,
   p_type            ,
   p_size            ,
   p_brand           ,
   p_name            ,
   p_container        ,
   p_mfgr            ,
   p_retailprice      ,
   p_comment
from part_et;
rem drop table part_et;
}
*wait
*sql
{
connect tpch/tpch;
set timing on
set echo on
!date

rem drop table customer;
create table customer(
   c_custkey          NOT NULL,
   c_mktsegment        ,
   c_nationkey         ,
   c_name            ,
   c_address          ,
   c_phone            ,
   c_acctbal          ,
   c_comment
)
pctfree 0
pctused 99
tablespace ts_c
parallel
storage (initial 80m freelists 84)
nologging
partition by hash (c_custkey)
partitions 128
as select
   c_custkey          ,
   c_mktsegment        ,
   c_nationkey         ,
   c_name            ,
   c_address          ,
   c_phone            ,
   c_acctbal          ,
   c_comment
from customer_et;
rem drop table customer_et;
}
*wait
*sql
{
connect tpch/tpch;
set timing on
set echo on
!date

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

```
   s_name            ,
   s_address          ,
   s_phone            ,
   s_acctbal
)
pctfree 0
pctused 99
tablespace ts_s
parallel
storage (initial 13m freelists 84)
nologging
partition by hash (s_suppkey)
partitions 128
as select
   s_suppkey          ,
   s_nationkey         ,
   s_comment          ,
   s_name            ,
   s_address          ,
   s_phone            ,
   s_acctbal
from supplier_et;
rem drop table supplier_et;
}
*wait
*sql
{
connect tpch/tpch;
set timing on
set echo on
!date

rem drop table nation;
create table nation(
   n_nationkey        NOT NULL,
   n_name            NOT NULL,
   n_regionkey        NOT NULL,
   n_comment          NOT NULL
)
tablespace ts_def
as select * from nation_et;
rem drop table nation_et;
}
*wait
*sql
{
connect tpch/tpch;
set timing on
set echo on
!date

rem drop table region;
create table region(
   r_regionkey        NOT NULL,
   r_name            NOT NULL,
   r_comment          NOT NULL
)
tablespace ts_def
as select * from region_et;
rem drop table region_et;
}
*wait
*sql
{
connect tpch/tpch;

drop table lineitem_et;
drop table orders_et;
drop table part_et;
drop table partsupp_et;
drop table supplier_et;
drop table customer_et;
drop table nation_et;
drop table region_et;
}
*wait
*wait
*bgoff
%e-dapop
%b-ixcre
*bgon=1
###########################################
##################################
# Index Creation Phase
*sql
{
connect tpch/tpch;

-- alter session force parallel dml parallel;
-- alter session set
isolation_level=serializable;

drop index l_orderkey;
create index l_orderkey
on lineitem (l_orderkey)
pctfree 2
initrans 10
compute statistics
tablespace ts_lokey
storage (freelist groups 4 freelists 84)
parallel
;
}
*sql
{
connect tpch/tpch;

-- alter session force parallel dml parallel;
-- alter session set
isolation_level=serializable;

drop index o_orderkey;
create unique index o_orderkey
on orders (o_orderkey)
pctfree 2
initrans 10
compute statistics
tablespace ts_okey
storage (freelist groups 4 freelists 84)
parallel
;
}
*sql
{
connect tpch/tpch;

-- alter session force parallel dml parallel;
-- alter session set
isolation_level=serializable;

drop index c_custkey;
create unique index c_custkey
on customer (c_custkey)
pctfree 2
initrans 10
compute statistics
tablespace ts_ckey
storage (freelists 84)
parallel
;
}
*wait
*sql
{
connect tpch/tpch;

alter index l_orderkey allocate extent (size
10000m instance 1);
alter index l_orderkey allocate extent (size
10000m instance 1);
alter index o_orderkey allocate extent (size
10000m instance 1);
}
*wait
*bgoff
%e-ixcre
%b-anlyz
*bgon=1
###########################################
##################################
# Analyze Phase
*sql
{
connect tpch/tpch;


execute
dbms_stats.gather_schema_stats('tpch' ,
estimate_percent => 1, degree => 128 ,
granularity => 'GLOBAL');
}
*wait
*wait
*bgoff
%e-anlyz
```

===
a_query.sql
===

```
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;
```

===
a_query2.sql
===

```
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;
```

===

atom.sh

===

```
#!/bin/ksh

. $KIT_DIR/env_mg

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

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

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

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

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

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

===

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)

*/

#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,OC
I_ATTR_SERVER,errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,lname,strl
en(lname),OCI_ATTR_USERNAME,
        errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,str
len(passwd),OCI_ATTR_PASSWORD,
        errhp);

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


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcusr,0,OC
I_ATTR_SESSION,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((c
har *)sqlstmt),OCI_NTV_SYNTAX,OCI_DEFAULT);
```

```
OCIbbname(curi,&l_keyi_bp,errhp,":l_key",AD
R(l_key),SIZ(l_key),SQLT_INT);

OCIbbname(curi,&o_keyi_bp,errhp,":o_key",AD
R(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),SQLT_INT);

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

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

OCIbbname(curr,l_pkey_bp,errhp,":l_pkey",AD
R(l_pkey),SIZ(l_pkey),SQLT_INT);

OCIbbname(curr,l_skey_bp,errhp,":l_skey",AD
R(l_skey),SIZ(l_skey),SQLT_INT);

OCIbbname(curr,l_quan_bp,errhp,":l_quan",AD
R(l_quan),SIZ(l_quan),SQLT_INT);

OCIbbname(curr,l_newquan_bp,errhp,":l_newqu
an",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",AD
R(l_disc),SIZ(l_disc),SQLT_FLT);

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

OCIbbname(curr,l_neweprice_bp,errhp,":l_new
eprice",ADR(l_neweprice),
          SIZ(l_neweprice),SQLT_FLT);


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

OCIbbname(curr,o_newtprice_bp,errhp,":o_new
tprice",ADR(o_newtprice),
          SIZ(o_newtprice), SQLT_FLT);

OCIbbname(curr,rprice_bp,errhp,":rprice",AD
R(rprice),SIZ(rprice), SQLT_FLT);

OCIbbname(curr,cost_bp,errhp,":cost",ADR(co
st),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_n
eweprice",ADR(l_neweprice),
          SIZ(l_neweprice),SQLT_FLT);

OCIbbname(cure1,l_newquan1_bp,errhp,":l_new
quan",ADR(l_newquan),
          SIZ(l_newquan),SQLT_INT);

OCIbbname(cure1,o_key1_bp,errhp,":o_key",AD
R(o_key),SIZ(o_key),SQLT_INT);

OCIbbname(cure1,l_key1_bp,errhp,":l_key",AD
R(l_key),SIZ(l_key),SQLT_INT);


OCIbbname(cure2,o_newtprice2_bp,errhp,":o_n
ewtprice",ADR(o_newtprice),
          SIZ(o_newtprice),SQLT_FLT);

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

}
```

===

atranspl.h

===

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

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

   DESCRIPTION

   MODIFIED   (MM/DD/YY)

*/
#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,it
er,0,NULL,NULL,OCI_DEFAULT)) != OCI_SUCCESS)
\
      sql_error(errh,status,1); \
   else \
      DISCARD 0

#define
OCIbbname(stmh,bindp,errh,sqlvar,progv,prog
vl,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,pro
gvl,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_DEF
AULT)) != OCI_SUCCESS) \
       sql_error(errh,status,1); \
    else \
       DISCARD 0

#define OCIrol(svcp,errh) \

if((status=OCITransRollback(svcp,errh,OCI_D
EFAULT)) != 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 4)"
#define OICATXT "alter session set
optimizer_index_cost_adj=25"

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

#define SQLTXT2 "BEGIN
d_atrans.doatrans(:l_key, :o_key, :delta, :
l_pkey, \
:l_skey, :l_quan, :l_newquan, :l_tax, :l_di
sc, :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 */
```

===

ckpt.sh

===

```
#!/bin/ksh

. $KIT_DIR/env_mg

sqlplus -s /NOLOG<< !

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

```
!
```

```
==========================================
```

cnt_hist.sql

```
==========================================


set verify off
set termout on
set echo on

select count(*)  from HISTORY;

select to_char(sysdate,'YYYY-MM-DD
HH:MI:SS') CURRENT_DATE from       dual;

exit;
```

```
==========================================
```

consist.sh

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

. $KIT_DIR/env_mg

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

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

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

STREAM=${NUM_STREAMS}
STREAM=8
let STREAM="$STREAM + 1" # add one for the
update stream
ITER=100
PROG=atranspl
USER=${DATABASE_USER}
CK=10

usage() {

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

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

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


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

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

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

i=0

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

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

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

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

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

i=0

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

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

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

wait

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

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

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

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

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

==

consist.sql

==

```
set verify off



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
```

==

cnt_commits.sh

==

```
if [ $# -lt 1 ]
then
echo 'USAGE:' $0 '<number of streams> '
exit
fi

. $KIT_DIR/env_mg

NS=$1
H=`expr $NS - 1`
i=0
sum=0
while [ $i -lt $NS ]
do
file=$ACID_OUT/dura$i
h=`fgrep -c Completed $file`
sum=`expr $sum + $h`
i=`expr $i + 1`
done

echo ' sum of completed TAs of success-files
($ACID_OUT/dura[0..'$H']): ' $sum
```

==

d_hist.sql

==

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

drop table history;

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

exit;
```

==

end_acid.sh

==

```
#!/bin/ksh

. $KIT_DIR/env_mg

OH=$ORACLE_HOME
OUT_DIR=$ACID_OUT/
DURA_DIR=$ACID_OUT/dura
RUN_ID_FILE=$ACID_DIR/run_id

ITER=10000
STEM=9
PROG=${ACID_DIR}/atranspl.ott
IN=${ACID_DIR}/acid_in
DURA=${ACID_OUT}/dura
OUT=${ACID_OUT}/drate
DSMPL=${DURA_DIR}/durasmpl
KEY=${DURA_DIR}/key${1}_
USER=tpch/tpch
TRIG=1
HCNT=duracnta


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


i=0
while [ $i -lt $STEM ]
do
   for j in `head -10 ${KEY}${i} | awk '{printf
"%d ",$1}'`
   do
      sqlplus $USER @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
echo "counts of entries in
successfile(dura"$i" )":>>$ACID_OUT/drate$i
fgrep -c Completed $ACID_OUT/dura$i >>
$ACID_OUT/drate$i
i=`expr $i + 1`
done
```

==

gettime.c

==

```
#ifdef RCSID
static char *RCSid =
#endif /* RCSID */

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

/*

   NAME
     gettime.c

   DESCRIPTION
     get wall clock time.
     get cpu time.

   FUNCTIONS
     get wall clock time.
     get cpu time.

   NOTES
     Both routines return time in seconds as a
double.
   MODIFIED   (MM/DD/YY)

*/


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

#define SUN_OS5

#if defined(SUN_OS5)
#define TIME_W_GETTIME
#define CPU_W_TIMES
#undef GETRU_STATS
#undef CPU_W_GETRU
#endif /* SUN_OS5 */

#if defined(sequent) || defined(SEQ_PSX)
# define GET_P_STATS
#endif /* sequent */

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

#if defined(a_osf) || defined(A_OSF)
# define TIME_W_GETTIME
# define CPU_W_GETRU
# define GETRU_STATS
#endif /* AIXRIOS */

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

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

#if !defined(CPU_W_GETRU)
&& !defined(CPU_W_TIMES)
# define CPU_W_TIMES
#endif

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

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

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

#if defined(TIME_W_TIMES) || defined
(CPU_W_TIMES)
# include <sys/types.h>
# include <sys/times.h>
# include <sys/param.h>  /* most systems
define HZ here */
#endif /* TIME_W_TIMES or CPU_W_TIMES */

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

# include <stdio.h>

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

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



double gettime ()

{

#ifdef TIME_W_GETTIME
   struct timeval tv;

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

#ifdef TIME_W_TIMES
   struct tms buf;

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

}



double getcpu ()

{

#ifdef CPU_W_TIMES
   struct tms buf;

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

#ifdef CPU_W_GETRU
   struct rusage ru;
   double usecs;

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

}



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

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

{

#ifdef GETRU_STATS
   struct rusage ru;

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

#ifdef GET_P_STATS
   timeval_t tv;
   struct process_stats ru;

   fprintf (fp, "%-10.10s %-10.10s %10d %10d ",
config,runname, proc_no, kids);
   if (kids)
      get_process_stats (&tv, PS_SELF, (struct
process_stats *) 0, &ru);
   else
```

```
    get_process_stats (&tv, PS_SELF, &ru,
(struct process_stats *) 0);
  print_ru (fp, &ru);
  fprintf (fp, "\n");
#endif /* GET_P_STATS */

}



getru1 (kids)

int kids;

{

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

#ifdef GET_P_STATS
  timeval_t tv;

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

}



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

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

{

#ifdef GETRU_STATS
  struct rusage ru;

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

#ifdef GET_P_STATS
  timeval_t tv;
  struct process_stats ru;

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

}



#ifdef GETRU_STATS

print_ru (fp, ru)

FILE *fp;
struct rusage *ru;

{

  fprintf (fp, "%10ld ", ru->ru_utime.tv_sec
* 1000 +

(ru->ru_utime.tv_usec/1000));
  fprintf (fp, "%10ld ", ru->ru_stime.tv_sec
* 1000 +

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

}



diffru (ru2, ru)

struct rusage *ru2;
struct rusage *ru;

{

  ru2->ru_utime.tv_sec -=
ru->ru_utime.tv_sec;
```

```
  ru2->ru_utime.tv_usec -=
ru->ru_utime.tv_usec;
  ru2->ru_stime.tv_sec -=
ru->ru_stime.tv_sec;
  ru2->ru_stime.tv_usec -=
ru->ru_stime.tv_usec;
  ru2->ru_maxrss -= ru->ru_maxrss;
  ru2->ru_ixrss -= ru->ru_ixrss;
  ru2->ru_idrss -= ru->ru_idrss;
  ru2->ru_minflt -= ru->ru_minflt;
  ru2->ru_majflt -= ru->ru_majflt;
  ru2->ru_nswap -= ru->ru_nswap;
  ru2->ru_inblock -= ru->ru_inblock;
  ru2->ru_oublock -= ru->ru_oublock;
  ru2->ru_msgsnd -= ru->ru_msgsnd;
  ru2->ru_msgrcv -= ru->ru_msgrcv;
  ru2->ru_nsignals -= ru->ru_nsignals;
  ru2->ru_nvcsw -= ru->ru_nvcsw;
  ru2->ru_nivcsw -= ru->ru_nivcsw;

}

#endif /* GETRU_STATS */



#ifdef GET_P_STATS

print_ru (fp, ps)

FILE *fp;
struct process_stats *ps;

{

  fprintf (fp, "%lu ", ps->ps_utime.tv_sec *
1000 +

(ps->ps_utime.tv_usec/1000));
  fprintf (fp, "%lu ", ps->ps_stime.tv_sec *
1000 +

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

}



diffru (ru2, ru)

struct process_stats *ru2;
struct process_stats *ru;

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

}

#endif /* GET_P_STATS */
```

===

iso1.sh

===

```
#!/bin/ksh
#       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.

. $KIT_DIR/env_mg

RSH=ksh

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


randkey 1 0.1 u"$USER" > $KEYFILE

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


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


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


sleep 10


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

===

iso2.sh

===

```
#!/bin/ksh

. $KIT_DIR/env_mg

RSH=ksh

OH=$ORACLE_HOME
OUT_DIR=$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



randkey 1 0.1 u"$USER" > $KEYFILE

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


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


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


sleep 10


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

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

===

iso3.sh

===

```
#!/bin/ksh
#       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.

. $KIT_DIR/env_mg

RSH=ksh

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



randkey 1 0.1 u"$USER" > $KEYFILE

sleep 1


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


sleep 10



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

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

===

iso4.sh

===

```
#!/bin/ksh

. $KIT_DIR/env_mg
```

```
RSH=ksh

OH=$ORACLE_HOME
OUT_DIR=$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



randkey 1 0.1 u"$USER" > $KEYFILE

sleep 1


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


sleep 10



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

wait

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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



===========================================
```

iso5.sh

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

. $KIT_DIR/env_mg

RSH=ksh

OH=$ORACLE_HOME
OUT_DIR=$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



randkey 1 0.1 u"$USER" > $KEYFILE

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


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

sleep 1


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


sleep 5


PSKEY=`randpsup 0.1`

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

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

wait

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

```
===========================================
```

iso6.sh

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

. $KIT_DIR/env_mg


RSH=ksh

QUERY_DIR=$KIT_DIR/queries_used
OH=/private/tpcd
OUT_DIR=$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



randkey 1 0.1 u"$USER" > $KEYFILE

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


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

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

sleep 1


#echo "Running Query 17b at `date`" >>
$TXN1FILE
#sqlplus $USER @a_q17b >> $TXN1FILE &
echo "Running Query 17b at `date`" >> $TXN1FILE
sqlplus $USER @$ACID_DIR/isolation/a_q17 >>
$TXN1FILE &


sleep 2


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

```
if [ "$HOST" != "" ]
then
echo "Starting ACID transaction on node $HOST"
>> $TXN2FILE
${RSH} -n ${HOST} $PROG 1 1 1 0 i$KEYFILE u$USER
s1 >> $TXN2FILE &
else
$PROG 1 1 1 0 i$KEYFILE u$USER s1 >> $TXN2FILE
&
fi


sleep 2

#echo "Running 2nd Query 17b at `date`" >>
$TXN3FILE
#sqlplus $USER @a_q17b >> $TXN3FILE &
echo "Running 2nd Query 17b at `date`" >>
$TXN3FILE
sqlplus $USER @$ACID_DIR/isolation/a_q17 >>
$TXN3FILE &

wait

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

cat $TXN1FILE $TXN2FILE $TXN3FILE >> $ISOFILE

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



===========================================
```

randkey.c

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

/*

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

   DESCRIPTION
     Generate random keys for ACID
transactions:
     O_ORDERKEY unique random (1..SF*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,OC
I_ATTR_SERVER,errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,lname,strl
en(lname),OCI_ATTR_USERNAME,
          errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,str
len(passwd),OCI_ATTR_PASSWORD,
          errhp);

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


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcusr,0,OC
I_ATTR_SESSION,errhp);

  /* Open and Parse cursor for query to choose
determine l_key. */
  /* Binds l_key to :l_key.
*/
  sprintf((char *) sqlstmt,SQLTXT1);
  OCIStmtPrepare(curi,errhp,(text
*)sqlstmt,strlen((char *)sqlstmt),
                 OCI_NTV_SYNTAX,OCI_DEFAULT);


OCIbbname(curi,l_key_bp,errhp,":l_key",ADR(
l_key),SIZ(l_key),SQLT_INT);

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

===

randpsup.c

===

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

/*

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

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

MODIFIED

*/

#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");
}
```

===

sample.sh

===

```
#!/bin/ksh


. $KIT_DIR/env_mg

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

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

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

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

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

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

===

sample.sql

===

```
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;
```

===

atrans.sql

===

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

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

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

```
IS
       ototal number;
       not_serializable EXCEPTION;
       PRAGMA
EXCEPTION_INIT(not_serializable,-8177);
BEGIN
  LOOP BEGIN

    select o_totalprice
          into o_tprice
          from orders
          where o_orderkey = o_key;

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

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

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

    update orders
       set o_totalprice = o_newtprice
       where o_orderkey = o_key;

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

    EXIT;

  EXCEPTION
    WHEN not_serializable THEN
      ROLLBACK;
  END;

  END LOOP;

END doatrans;
END;
/

exit;
```

===

run_acid.sh

===

```
#!/bin/ksh

. $KIT_DIR/env_mg

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 tpcd/tpcd"
   echo "-t trigger  : trigger time between
process starts, default is 1 second"
   echo "-h          : print this usage summary"
   exit 1;
}

ITER=10000
STEM=9
SF=1
PROG=atranspl
IN=${ACID_DIR}/acid_in
DURA_DIR=$ACID_OUT/dura
OUT=$ACID_OUT/drate
DURA=$ACID_OUT/dura
KEY=${DURA_DIR}/key$$_
USER=tpch/tpch
TRIG=1
HCNT=duracntb

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


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`


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

while [ $i -lt $STEM ]
do
   randkey 1000 ${SF} u${USER} > ${KEY}${i} &
   i=`expr $i + 1`
done
sync;

wait

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

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"
```

=====================================

Disk Configuration Details

=====================================

The following is the output of vxprint -th for a datafile volume as a sample (see chapter 5.2).

```
Disk group: dgrp1

V  NAME        RVG         KSTATE   STATE
LENGTH   READPOL   PREFPLEX UTYPE
PL NAME        VOLUME      KSTATE   STATE
LENGTH   LAYOUT    NCOL/WID MODE
SD NAME        PLEX        DISK     DISKOFFS
LENGTH   [COL/]OFF DEVICE   MODE
SV NAME        PLEX        VOLNAME  NVOLLAYR
LENGTH   [COL/]OFF AM/NM    MODE
DC NAME        PARENTVOL    LOGVOL
SP NAME        SNAPVOL     DCO

v  volline1-00  -          ENABLED  ACTIVE
30670848 ROUND     -       gen
pl line1-00    volline1-00  ENABLED  ACTIVE
30670848 STRIPE    288/2048 RW
sd c117t0d0-02  line1-00    c117t0d0 443072
106496   0/0      c117t0d0 ENA
sd c122t0d0-02  line1-00    c122t0d0 443072
106496   1/0      c122t0d0 ENA
sd c129t0d0-02  line1-00    c129t0d0 443072
106496   2/0      c129t0d0 ENA
sd c130t0d0-02  line1-00    c130t0d0 443072
106496   3/0      c130t0d0 ENA
sd c116t0d0-02  line1-00    c116t0d0 443072
106496   4/0      c116t0d0 ENA
sd c143t0d0-02  line1-00    c143t0d0 443072
106496   5/0      c143t0d0 ENA
sd c144t0d0-02  line1-00    c144t0d0 443072
106496   6/0      c144t0d0 ENA
sd c150t0d0-02  line1-00    c150t0d0 443072
106496   7/0      c150t0d0 ENA
sd c159t0d0-02  line1-00    c159t0d0 443072
106496   8/0      c159t0d0 ENA
sd c160t0d0-02  line1-00    c160t0d0 443072
106496   9/0      c160t0d0 ENA
sd c166t0d0-02  line1-00    c166t0d0 443072
106496   10/0     c166t0d0 ENA
sd c175t0d0-02  line1-00    c175t0d0 443072
106496   11/0     c175t0d0 ENA
sd c120t0d0-02  line1-00    c120t0d0 443072
106496   12/0     c120t0d0 ENA
sd c125t0d0-02  line1-00    c125t0d0 443072
106496   13/0     c125t0d0 ENA
sd c132t0d0-02  line1-00    c132t0d0 443072
106496   14/0     c132t0d0 ENA
sd c133t0d0-02  line1-00    c133t0d0 443072
106496   15/0     c133t0d0 ENA
sd c138t0d0-02  line1-00    c138t0d0 443072
106496   16/0     c138t0d0 ENA
sd c147t0d0-02  line1-00    c147t0d0 443072
106496   17/0     c147t0d0 ENA
sd c148t0d0-02  line1-00    c148t0d0 443072
106496   18/0     c148t0d0 ENA
sd c154t0d0-02  line1-00    c154t0d0 443072
106496   19/0     c154t0d0 ENA
sd c163t0d0-02  line1-00    c163t0d0 443072
106496   20/0     c163t0d0 ENA
sd c164t0d0-02  line1-00    c164t0d0 443072
106496   21/0     c164t0d0 ENA
sd c170t0d0-02  line1-00    c170t0d0 443072
106496   22/0     c170t0d0 ENA
sd c112t0d0-02  line1-00    c112t0d0 443072
106496   23/0     c112t0d0 ENA
sd c123t0d0-02  line1-00    c123t0d0 443072
106496   24/0     c123t0d0 ENA
sd c128t0d0-02  line1-00    c128t0d0 443072
106496   25/0     c128t0d0 ENA
sd c135t0d0-02  line1-00    c135t0d0 443072
106496   26/0     c135t0d0 ENA
sd c136t0d0-02  line1-00    c136t0d0 443072
106496   27/0     c136t0d0 ENA
sd c142t0d0-02  line1-00    c142t0d0 443072
106496   28/0     c142t0d0 ENA
sd c151t0d0-02  line1-00    c151t0d0 443072
106496   29/0     c151t0d0 ENA
sd c152t0d0-02  line1-00    c152t0d0 443072
106496   30/0     c152t0d0 ENA
sd c158t0d0-02  line1-00    c158t0d0 443072
106496   31/0     c158t0d0 ENA
sd c167t0d0-02  line1-00    c167t0d0 443072
106496   32/0     c167t0d0 ENA
sd c168t0d0-02  line1-00    c168t0d0 443072
106496   33/0     c168t0d0 ENA
```

```
sd c174t0d0-02  line1-00    c174t0d0 443072
106496   34/0     c174t0d0 ENA
sd c113t0d0-02  line1-00    c113t0d0 443072
106496   35/0     c113t0d0 ENA
sd c114t0d0-02  line1-00    c114t0d0 443072
106496   36/0     c114t0d0 ENA
sd c115t0d0-02  line1-00    c115t0d0 443072
106496   37/0     c115t0d0 ENA
sd c139t0d0-02  line1-00    c139t0d0 443072
106496   38/0     c139t0d0 ENA
sd c140t0d0-02  line1-00    c140t0d0 443072
106496   39/0     c140t0d0 ENA
sd c146t0d0-02  line1-00    c146t0d0 443072
106496   40/0     c146t0d0 ENA
sd c155t0d0-02  line1-00    c155t0d0 443072
106496   41/0     c155t0d0 ENA
sd c156t0d0-02  line1-00    c156t0d0 443072
106496   42/0     c156t0d0 ENA
sd c162t0d0-02  line1-00    c162t0d0 443072
106496   43/0     c162t0d0 ENA
sd c171t0d0-02  line1-00    c171t0d0 443072
106496   44/0     c171t0d0 ENA
sd c172t0d0-02  line1-00    c172t0d0 443072
106496   45/0     c172t0d0 ENA
sd c119t0d0-02  line1-00    c119t0d0 443072
106496   46/0     c119t0d0 ENA
sd c126t0d0-02  line1-00    c126t0d0 443072
106496   47/0     c126t0d0 ENA
sd c117t0d1-02  line1-00    c117t0d1 443072
106496   48/0     c117t0d1 ENA
sd c122t0d1-02  line1-00    c122t0d1 443072
106496   49/0     c122t0d1 ENA
sd c129t0d1-02  line1-00    c129t0d1 443072
106496   50/0     c129t0d1 ENA
sd c130t0d1-02  line1-00    c130t0d1 443072
106496   51/0     c130t0d1 ENA
sd c116t0d1-02  line1-00    c116t0d1 443072
106496   52/0     c116t0d1 ENA
sd c143t0d1-02  line1-00    c143t0d1 443072
106496   53/0     c143t0d1 ENA
sd c144t0d1-02  line1-00    c144t0d1 443072
106496   54/0     c144t0d1 ENA
sd c150t0d1-02  line1-00    c150t0d1 443072
106496   55/0     c150t0d1 ENA
sd c159t0d1-02  line1-00    c159t0d1 443072
106496   56/0     c159t0d1 ENA
sd c160t0d1-02  line1-00    c160t0d1 443072
106496   57/0     c160t0d1 ENA
sd c166t0d1-02  line1-00    c166t0d1 443072
106496   58/0     c166t0d1 ENA
sd c175t0d1-02  line1-00    c175t0d1 443072
106496   59/0     c175t0d1 ENA
sd c120t0d1-02  line1-00    c120t0d1 443072
106496   60/0     c120t0d1 ENA
sd c125t0d1-02  line1-00    c125t0d1 443072
106496   61/0     c125t0d1 ENA
sd c132t0d1-02  line1-00    c132t0d1 443072
106496   62/0     c132t0d1 ENA
sd c133t0d1-02  line1-00    c133t0d1 443072
106496   63/0     c133t0d1 ENA
sd c138t0d1-02  line1-00    c138t0d1 443072
106496   64/0     c138t0d1 ENA
sd c147t0d1-02  line1-00    c147t0d1 443072
106496   65/0     c147t0d1 ENA
sd c148t0d1-02  line1-00    c148t0d1 443072
106496   66/0     c148t0d1 ENA
sd c154t0d1-02  line1-00    c154t0d1 443072
106496   67/0     c154t0d1 ENA
sd c163t0d1-02  line1-00    c163t0d1 443072
106496   68/0     c163t0d1 ENA
sd c164t0d1-02  line1-00    c164t0d1 443072
106496   69/0     c164t0d1 ENA
```

```
sd c170t0d1-02  line1-00    c170t0d1 443072
106496   70/0     c170t0d1 ENA
sd c112t0d1-02  line1-00    c112t0d1 443072
106496   71/0     c112t0d1 ENA
sd c123t0d1-02  line1-00    c123t0d1 443072
106496   72/0     c123t0d1 ENA
sd c128t0d1-02  line1-00    c128t0d1 443072
106496   73/0     c128t0d1 ENA
sd c135t0d1-02  line1-00    c135t0d1 443072
106496   74/0     c135t0d1 ENA
sd c136t0d1-02  line1-00    c136t0d1 443072
106496   75/0     c136t0d1 ENA
sd c142t0d1-02  line1-00    c142t0d1 443072
106496   76/0     c142t0d1 ENA
sd c151t0d1-02  line1-00    c151t0d1 443072
106496   77/0     c151t0d1 ENA
sd c152t0d1-02  line1-00    c152t0d1 443072
106496   78/0     c152t0d1 ENA
sd c158t0d1-02  line1-00    c158t0d1 443072
106496   79/0     c158t0d1 ENA
sd c167t0d1-02  line1-00    c167t0d1 443072
106496   80/0     c167t0d1 ENA
sd c168t0d1-02  line1-00    c168t0d1 443072
106496   81/0     c168t0d1 ENA
sd c174t0d1-02  line1-00    c174t0d1 443072
106496   82/0     c174t0d1 ENA
sd c113t0d1-02  line1-00    c113t0d1 443072
106496   83/0     c113t0d1 ENA
sd c114t0d1-02  line1-00    c114t0d1 443072
106496   84/0     c114t0d1 ENA
sd c115t0d1-02  line1-00    c115t0d1 443072
106496   85/0     c115t0d1 ENA
sd c139t0d1-02  line1-00    c139t0d1 443072
106496   86/0     c139t0d1 ENA
sd c140t0d1-02  line1-00    c140t0d1 443072
106496   87/0     c140t0d1 ENA
sd c146t0d1-02  line1-00    c146t0d1 443072
106496   88/0     c146t0d1 ENA
sd c155t0d1-02  line1-00    c155t0d1 443072
106496   89/0     c155t0d1 ENA
sd c156t0d1-02  line1-00    c156t0d1 443072
106496   90/0     c156t0d1 ENA
sd c162t0d1-02  line1-00    c162t0d1 443072
106496   91/0     c162t0d1 ENA
sd c171t0d1-02  line1-00    c171t0d1 443072
106496   92/0     c171t0d1 ENA
sd c172t0d1-02  line1-00    c172t0d1 443072
106496   93/0     c172t0d1 ENA
sd c119t0d1-02  line1-00    c119t0d1 443072
106496   94/0     c119t0d1 ENA
sd c126t0d1-02  line1-00    c126t0d1 443072
106496   95/0     c126t0d1 ENA
sd c117t0d2-02  line1-00    c117t0d2 443072
106496   96/0     c117t0d2 ENA
sd c122t0d2-02  line1-00    c122t0d2 443072
106496   97/0     c122t0d2 ENA
sd c129t0d2-02  line1-00    c129t0d2 443072
106496   98/0     c129t0d2 ENA
sd c130t0d2-02  line1-00    c130t0d2 443072
106496   99/0     c130t0d2 ENA
sd c116t0d2-02  line1-00    c116t0d2 443072
106496   100/0    c116t0d2 ENA
sd c143t0d2-02  line1-00    c143t0d2 443072
106496   101/0    c143t0d2 ENA
sd c144t0d2-02  line1-00    c144t0d2 443072
106496   102/0    c144t0d2 ENA
sd c150t0d2-02  line1-00    c150t0d2 443072
106496   103/0    c150t0d2 ENA
sd c159t0d2-02  line1-00    c159t0d2 443072
106496   104/0    c159t0d2 ENA
sd c160t0d2-02  line1-00    c160t0d2 443072
106496   105/0    c160t0d2 ENA
```

```
sd c166t0d2-02  line1-00    c166t0d2 443072
106496   106/0    c166t0d2 ENA
sd c175t0d2-02  line1-00    c175t0d2 443072
106496   107/0    c175t0d2 ENA
sd c120t0d2-02  line1-00    c120t0d2 443072
106496   108/0    c120t0d2 ENA
sd c125t0d2-02  line1-00    c125t0d2 443072
106496   109/0    c125t0d2 ENA
sd c132t0d2-02  line1-00    c132t0d2 443072
106496   110/0    c132t0d2 ENA
sd c133t0d2-02  line1-00    c133t0d2 443072
106496   111/0    c133t0d2 ENA
sd c138t0d2-02  line1-00    c138t0d2 443072
106496   112/0    c138t0d2 ENA
sd c147t0d2-02  line1-00    c147t0d2 443072
106496   113/0    c147t0d2 ENA
sd c148t0d2-02  line1-00    c148t0d2 443072
106496   114/0    c148t0d2 ENA
sd c154t0d2-02  line1-00    c154t0d2 443072
106496   115/0    c154t0d2 ENA
sd c163t0d2-02  line1-00    c163t0d2 443072
106496   116/0    c163t0d2 ENA
sd c164t0d2-02  line1-00    c164t0d2 443072
106496   117/0    c164t0d2 ENA
sd c170t0d2-02  line1-00    c170t0d2 443072
106496   118/0    c170t0d2 ENA
sd c112t0d2-02  line1-00    c112t0d2 443072
106496   119/0    c112t0d2 ENA
sd c123t0d2-02  line1-00    c123t0d2 443072
106496   120/0    c123t0d2 ENA
sd c128t0d2-02  line1-00    c128t0d2 443072
106496   121/0    c128t0d2 ENA
sd c135t0d2-02  line1-00    c135t0d2 443072
106496   122/0    c135t0d2 ENA
sd c136t0d2-02  line1-00    c136t0d2 443072
106496   123/0    c136t0d2 ENA
sd c142t0d2-02  line1-00    c142t0d2 443072
106496   124/0    c142t0d2 ENA
sd c151t0d2-02  line1-00    c151t0d2 443072
106496   125/0    c151t0d2 ENA
sd c152t0d2-02  line1-00    c152t0d2 443072
106496   126/0    c152t0d2 ENA
sd c158t0d2-02  line1-00    c158t0d2 443072
106496   127/0    c158t0d2 ENA
sd c167t0d2-02  line1-00    c167t0d2 443072
106496   128/0    c167t0d2 ENA
sd c168t0d2-02  line1-00    c168t0d2 443072
106496   129/0    c168t0d2 ENA
sd c174t0d2-02  line1-00    c174t0d2 443072
106496   130/0    c174t0d2 ENA
sd c113t0d2-02  line1-00    c113t0d2 443072
106496   131/0    c113t0d2 ENA
sd c114t0d2-02  line1-00    c114t0d2 443072
106496   132/0    c114t0d2 ENA
sd c115t0d2-02  line1-00    c115t0d2 443072
106496   133/0    c115t0d2 ENA
sd c139t0d2-02  line1-00    c139t0d2 443072
106496   134/0    c139t0d2 ENA
sd c140t0d2-02  line1-00    c140t0d2 443072
106496   135/0    c140t0d2 ENA
sd c146t0d2-02  line1-00    c146t0d2 443072
106496   136/0    c146t0d2 ENA
sd c155t0d2-02  line1-00    c155t0d2 443072
106496   137/0    c155t0d2 ENA
sd c156t0d2-02  line1-00    c156t0d2 443072
106496   138/0    c156t0d2 ENA
sd c162t0d2-02  line1-00    c162t0d2 443072
106496   139/0    c162t0d2 ENA
sd c171t0d2-02  line1-00    c171t0d2 443072
106496   140/0    c171t0d2 ENA
sd c172t0d2-02  line1-00    c172t0d2 443072
106496   141/0    c172t0d2 ENA
```

```
sd c119t0d2-02  line1-00    c119t0d2 443072
106496   142/0    c119t0d2 ENA
sd c126t0d2-02  line1-00    c126t0d2 443072
106496   143/0    c126t0d2 ENA
sd c117t0d3-02  line1-00    c117t0d3 443072
106496   144/0    c117t0d3 ENA
sd c122t0d3-02  line1-00    c122t0d3 443072
106496   145/0    c122t0d3 ENA
sd c129t0d3-02  line1-00    c129t0d3 443072
106496   146/0    c129t0d3 ENA
sd c130t0d3-02  line1-00    c130t0d3 443072
106496   147/0    c130t0d3 ENA
sd c116t0d3-02  line1-00    c116t0d3 443072
106496   148/0    c116t0d3 ENA
sd c143t0d3-02  line1-00    c143t0d3 443072
106496   149/0    c143t0d3 ENA
sd c144t0d3-02  line1-00    c144t0d3 443072
106496   150/0    c144t0d3 ENA
sd c150t0d3-02  line1-00    c150t0d3 443072
106496   151/0    c150t0d3 ENA
sd c159t0d3-02  line1-00    c159t0d3 443072
106496   152/0    c159t0d3 ENA
sd c160t0d3-02  line1-00    c160t0d3 443072
106496   153/0    c160t0d3 ENA
sd c166t0d3-02  line1-00    c166t0d3 443072
106496   154/0    c166t0d3 ENA
sd c175t0d3-02  line1-00    c175t0d3 443072
106496   155/0    c175t0d3 ENA
sd c120t0d3-02  line1-00    c120t0d3 443072
106496   156/0    c120t0d3 ENA
sd c125t0d3-02  line1-00    c125t0d3 443072
106496   157/0    c125t0d3 ENA
sd c132t0d3-02  line1-00    c132t0d3 443072
106496   158/0    c132t0d3 ENA
sd c133t0d3-02  line1-00    c133t0d3 443072
106496   159/0    c133t0d3 ENA
sd c138t0d3-02  line1-00    c138t0d3 443072
106496   160/0    c138t0d3 ENA
sd c147t0d3-02  line1-00    c147t0d3 443072
106496   161/0    c147t0d3 ENA
sd c148t0d3-02  line1-00    c148t0d3 443072
106496   162/0    c148t0d3 ENA
sd c154t0d3-02  line1-00    c154t0d3 443072
106496   163/0    c154t0d3 ENA
sd c163t0d3-02  line1-00    c163t0d3 443072
106496   164/0    c163t0d3 ENA
sd c164t0d3-02  line1-00    c164t0d3 443072
106496   165/0    c164t0d3 ENA
sd c170t0d3-02  line1-00    c170t0d3 443072
106496   166/0    c170t0d3 ENA
sd c112t0d3-02  line1-00    c112t0d3 443072
106496   167/0    c112t0d3 ENA
sd c123t0d3-02  line1-00    c123t0d3 443072
106496   168/0    c123t0d3 ENA
sd c128t0d3-02  line1-00    c128t0d3 443072
106496   169/0    c128t0d3 ENA
sd c135t0d3-02  line1-00    c135t0d3 443072
106496   170/0    c135t0d3 ENA
sd c136t0d3-02  line1-00    c136t0d3 443072
106496   171/0    c136t0d3 ENA
sd c142t0d3-02  line1-00    c142t0d3 443072
106496   172/0    c142t0d3 ENA
sd c151t0d3-02  line1-00    c151t0d3 443072
106496   173/0    c151t0d3 ENA
sd c152t0d3-02  line1-00    c152t0d3 443072
106496   174/0    c152t0d3 ENA
sd c158t0d3-02  line1-00    c158t0d3 443072
106496   175/0    c158t0d3 ENA
sd c167t0d3-02  line1-00    c167t0d3 443072
106496   176/0    c167t0d3 ENA
sd c168t0d3-02  line1-00    c168t0d3 443072
106496   177/0    c168t0d3 ENA
```

```
sd c174t0d3-02  line1-00    c174t0d3 443072
106496  178/0    c174t0d3 ENA
sd c113t0d3-02  line1-00    c113t0d3 443072
106496  179/0    c113t0d3 ENA
sd c114t0d3-02  line1-00    c114t0d3 443072
106496  180/0    c114t0d3 ENA
sd c115t0d3-02  line1-00    c115t0d3 443072
106496  181/0    c115t0d3 ENA
sd c139t0d3-02  line1-00    c139t0d3 443072
106496  182/0    c139t0d3 ENA
sd c140t0d3-02  line1-00    c140t0d3 443072
106496  183/0    c140t0d3 ENA
sd c146t0d3-02  line1-00    c146t0d3 443072
106496  184/0    c146t0d3 ENA
sd c155t0d3-02  line1-00    c155t0d3 443072
106496  185/0    c155t0d3 ENA
sd c156t0d3-02  line1-00    c156t0d3 443072
106496  186/0    c156t0d3 ENA
sd c162t0d3-02  line1-00    c162t0d3 443072
106496  187/0    c162t0d3 ENA
sd c171t0d3-02  line1-00    c171t0d3 443072
106496  188/0    c171t0d3 ENA
sd c172t0d3-02  line1-00    c172t0d3 443072
106496  189/0    c172t0d3 ENA
sd c119t0d3-02  line1-00    c119t0d3 443072
106496  190/0    c119t0d3 ENA
sd c126t0d3-02  line1-00    c126t0d3 443072
106496  191/0    c126t0d3 ENA
sd c117t0d4-02  line1-00    c117t0d4 443072
106496  192/0    c117t0d4 ENA
sd c122t0d4-02  line1-00    c122t0d4 443072
106496  193/0    c122t0d4 ENA
sd c129t0d4-02  line1-00    c129t0d4 443072
106496  194/0    c129t0d4 ENA
sd c130t0d4-02  line1-00    c130t0d4 443072
106496  195/0    c130t0d4 ENA
sd c116t0d4-02  line1-00    c116t0d4 443072
106496  196/0    c116t0d4 ENA
sd c143t0d4-02  line1-00    c143t0d4 443072
106496  197/0    c143t0d4 ENA
sd c144t0d4-02  line1-00    c144t0d4 443072
106496  198/0    c144t0d4 ENA
sd c150t0d4-02  line1-00    c150t0d4 443072
106496  199/0    c150t0d4 ENA
sd c159t0d4-02  line1-00    c159t0d4 443072
106496  200/0    c159t0d4 ENA
sd c160t0d4-02  line1-00    c160t0d4 443072
106496  201/0    c160t0d4 ENA
sd c166t0d4-02  line1-00    c166t0d4 443072
106496  202/0    c166t0d4 ENA
sd c175t0d4-02  line1-00    c175t0d4 443072
106496  203/0    c175t0d4 ENA
sd c120t0d4-02  line1-00    c120t0d4 443072
106496  204/0    c120t0d4 ENA
sd c125t0d4-02  line1-00    c125t0d4 443072
106496  205/0    c125t0d4 ENA
sd c132t0d4-02  line1-00    c132t0d4 443072
106496  206/0    c132t0d4 ENA
sd c133t0d4-02  line1-00    c133t0d4 443072
106496  207/0    c133t0d4 ENA
sd c138t0d4-02  line1-00    c138t0d4 443072
106496  208/0    c138t0d4 ENA
sd c147t0d4-02  line1-00    c147t0d4 443072
106496  209/0    c147t0d4 ENA
sd c148t0d4-02  line1-00    c148t0d4 443072
106496  210/0    c148t0d4 ENA
sd c154t0d4-02  line1-00    c154t0d4 443072
106496  211/0    c154t0d4 ENA
sd c163t0d4-02  line1-00    c163t0d4 443072
106496  212/0    c163t0d4 ENA
sd c164t0d4-02  line1-00    c164t0d4 443072
106496  213/0    c164t0d4 ENA
sd c170t0d4-02  line1-00    c170t0d4 443072
106496  214/0    c170t0d4 ENA
sd c112t0d4-02  line1-00    c112t0d4 443072
106496  215/0    c112t0d4 ENA
sd c123t0d4-02  line1-00    c123t0d4 443072
106496  216/0    c123t0d4 ENA
sd c128t0d4-02  line1-00    c128t0d4 443072
106496  217/0    c128t0d4 ENA
sd c135t0d4-02  line1-00    c135t0d4 443072
106496  218/0    c135t0d4 ENA
sd c136t0d4-02  line1-00    c136t0d4 443072
106496  219/0    c136t0d4 ENA
sd c142t0d4-02  line1-00    c142t0d4 443072
106496  220/0    c142t0d4 ENA
sd c151t0d4-02  line1-00    c151t0d4 443072
106496  221/0    c151t0d4 ENA
sd c152t0d4-02  line1-00    c152t0d4 443072
106496  222/0    c152t0d4 ENA
sd c158t0d4-02  line1-00    c158t0d4 443072
106496  223/0    c158t0d4 ENA
sd c167t0d4-02  line1-00    c167t0d4 443072
106496  224/0    c167t0d4 ENA
sd c168t0d4-02  line1-00    c168t0d4 443072
106496  225/0    c168t0d4 ENA
sd c174t0d4-02  line1-00    c174t0d4 443072
106496  226/0    c174t0d4 ENA
sd c113t0d4-02  line1-00    c113t0d4 443072
106496  227/0    c113t0d4 ENA
sd c114t0d4-02  line1-00    c114t0d4 443072
106496  228/0    c114t0d4 ENA
sd c115t0d4-02  line1-00    c115t0d4 443072
106496  229/0    c115t0d4 ENA
sd c139t0d4-02  line1-00    c139t0d4 443072
106496  230/0    c139t0d4 ENA
sd c140t0d4-02  line1-00    c140t0d4 443072
106496  231/0    c140t0d4 ENA
sd c146t0d4-02  line1-00    c146t0d4 443072
106496  232/0    c146t0d4 ENA
sd c155t0d4-02  line1-00    c155t0d4 443072
106496  233/0    c155t0d4 ENA
sd c156t0d4-02  line1-00    c156t0d4 443072
106496  234/0    c156t0d4 ENA
sd c162t0d4-02  line1-00    c162t0d4 443072
106496  235/0    c162t0d4 ENA
sd c171t0d4-02  line1-00    c171t0d4 443072
106496  236/0    c171t0d4 ENA
sd c172t0d4-02  line1-00    c172t0d4 443072
106496  237/0    c172t0d4 ENA
sd c119t0d4-02  line1-00    c119t0d4 443072
106496  238/0    c119t0d4 ENA
sd c126t0d4-02  line1-00    c126t0d4 443072
106496  239/0    c126t0d4 ENA
sd c118t0d0-02  line1-00    c118t0d0 443072
106496  240/0    c118t0d0 ENA
sd c121t0d0-02  line1-00    c121t0d0 443072
106496  241/0    c121t0d0 ENA
sd c124t0d0-02  line1-00    c124t0d0 443072
106496  242/0    c124t0d0 ENA
sd c127t0d0-02  line1-00    c127t0d0 443072
106496  243/0    c127t0d0 ENA
sd c131t0d0-02  line1-00    c131t0d0 443072
106496  244/0    c131t0d0 ENA
sd c134t0d0-02  line1-00    c134t0d0 443072
106496  245/0    c134t0d0 ENA
sd c137t0d0-02  line1-00    c137t0d0 443072
106496  246/0    c137t0d0 ENA
sd c141t0d0-02  line1-00    c141t0d0 443072
106496  247/0    c141t0d0 ENA
sd c145t0d0-02  line1-00    c145t0d0 443072
106496  248/0    c145t0d0 ENA
sd c149t0d0-02  line1-00    c149t0d0 443072
106496  249/0    c149t0d0 ENA
```

```
sd c153t0d0-02  line1-00    c153t0d0 443072
106496   250/0    c153t0d0 ENA
sd c157t0d0-02  line1-00    c157t0d0 443072
106496   251/0    c157t0d0 ENA
sd c161t0d0-02  line1-00    c161t0d0 443072
106496   252/0    c161t0d0 ENA
sd c165t0d0-02  line1-00    c165t0d0 443072
106496   253/0    c165t0d0 ENA
sd c169t0d0-02  line1-00    c169t0d0 443072
106496   254/0    c169t0d0 ENA
sd c173t0d0-02  line1-00    c173t0d0 443072
106496   255/0    c173t0d0 ENA
sd c118t0d1-02  line1-00    c118t0d1 443072
106496   256/0    c118t0d1 ENA
sd c121t0d1-02  line1-00    c121t0d1 443072
106496   257/0    c121t0d1 ENA
sd c124t0d1-02  line1-00    c124t0d1 443072
106496   258/0    c124t0d1 ENA
sd c127t0d1-02  line1-00    c127t0d1 443072
106496   259/0    c127t0d1 ENA
sd c131t0d1-02  line1-00    c131t0d1 443072
106496   260/0    c131t0d1 ENA
sd c134t0d1-02  line1-00    c134t0d1 443072
106496   261/0    c134t0d1 ENA
sd c137t0d1-02  line1-00    c137t0d1 443072
106496   262/0    c137t0d1 ENA
sd c141t0d1-02  line1-00    c141t0d1 443072
106496   263/0    c141t0d1 ENA
sd c145t0d1-02  line1-00    c145t0d1 443072
106496   264/0    c145t0d1 ENA
sd c149t0d1-02  line1-00    c149t0d1 443072
106496   265/0    c149t0d1 ENA
sd c153t0d1-02  line1-00    c153t0d1 443072
106496   266/0    c153t0d1 ENA
sd c157t0d1-02  line1-00    c157t0d1 443072
106496   267/0    c157t0d1 ENA
sd c161t0d1-02  line1-00    c161t0d1 443072
106496   268/0    c161t0d1 ENA
sd c165t0d1-02  line1-00    c165t0d1 443072
106496   269/0    c165t0d1 ENA
sd c169t0d1-02  line1-00    c169t0d1 443072
106496   270/0    c169t0d1 ENA
sd c173t0d1-02  line1-00    c173t0d1 443072
106496   271/0    c173t0d1 ENA
sd c118t0d2-02  line1-00    c118t0d2 443072
106496   272/0    c118t0d2 ENA
sd c121t0d2-02  line1-00    c121t0d2 443072
106496   273/0    c121t0d2 ENA
sd c124t0d2-02  line1-00    c124t0d2 443072
106496   274/0    c124t0d2 ENA
sd c127t0d2-02  line1-00    c127t0d2 443072
106496   275/0    c127t0d2 ENA
sd c131t0d2-02  line1-00    c131t0d2 443072
106496   276/0    c131t0d2 ENA
sd c134t0d2-02  line1-00    c134t0d2 443072
106496   277/0    c134t0d2 ENA
sd c137t0d2-02  line1-00    c137t0d2 443072
106496   278/0    c137t0d2 ENA
sd c141t0d2-02  line1-00    c141t0d2 443072
106496   279/0    c141t0d2 ENA
sd c145t0d2-02  line1-00    c145t0d2 443072
106496   280/0    c145t0d2 ENA
sd c149t0d2-02  line1-00    c149t0d2 443072
106496   281/0    c149t0d2 ENA
sd c153t0d2-02  line1-00    c153t0d2 443072
106496   282/0    c153t0d2 ENA
sd c157t0d2-02  line1-00    c157t0d2 443072
106496   283/0    c157t0d2 ENA
sd c161t0d2-02  line1-00    c161t0d2 443072
106496   284/0    c161t0d2 ENA
sd c165t0d2-02  line1-00    c165t0d2 443072
106496   285/0    c165t0d2 ENA
sd c169t0d2-02  line1-00    c169t0d2 443072
106496   286/0    c169t0d2 ENA
sd c173t0d2-02  line1-00    c173t0d2 443072
106496   287/0    c173t0d2 ENA
```

Appendix C. Query Text and Query Output

```
========================================
```

qual1.v1

```
========================================
Begin Execution at Tue Aug 19 12:31:52 2003

-- using default substitutions

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

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

4 rows processed.
Statement Processed in 1.07 seconds.

Ended Executing this Query at Tue Aug 19 12:31:53
2003

Query Started at 1061289112.87
Query Ended at 1061289113.94
Query Processed in 1.07 seconds

SQL statements processed: 1
Queries processed: 1

========================================
```

qual2.v1

```
========================================
Begin Execution at Tue Aug 19 12:31:54 2003

-- using default substitutions

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

S_ACCTBAL                    S_NAME
N_NAME
P_PARTKEY                    P_MFGR
S_ADDRESS
S_PHONE
S_COMMENT
9938.53                   Supplier#000005359
UNITED KINGDOM
185358.00                  Manufacturer#4
QKuHYh,vZGiwu2FWEJoLDx04
33-429-790-6131
blithely silent pinto beans are furiously. slyly final
deposits acros
9937.84                   Supplier#000005969
ROMANIA
108438.00                  Manufacturer#1
ANDENSOSmk,miq23Xfb5RWt6dvUcvt6Qa
29-520-692-3537
carefully slow deposits use furiously. slyly ironic
platelets above the ironic
9936.22                   Supplier#000005250
UNITED KINGDOM
249.00                    Manufacturer#4
B3rqp0xbSEim4Mpy2RH J
33-320-228-2957
blithely special packages are. stealthily express
deposits across the closely final instructi
9923.77                   Supplier#000002324
GERMANY
29821.00                   Manufacturer#4

... rows truncated ...

125988.00                  Manufacturer#2
riRcntps4KEDtYScjpMIWeYF6mNnR
32-194-698-3365
final, ironic theodolites alongside of the ironic
7912.91                   Supplier#000004211
GERMANY
159180.00                  Manufacturer#5
2wQRVovHrm3,v03IKzfTd,1PYsFXQFFOG
17-266-947-7315
final requests integrate slyly above the silent, even
7912.91                   Supplier#000004211
GERMANY
184210.00                  Manufacturer#4
2wQRVovHrm3,v03IKzfTd,1PYsFXQFFOG
17-266-947-7315
final requests integrate slyly above the silent, even
7894.56                   Supplier#000007981
GERMANY
85472.00                   Manufacturer#4
NSJ96vMROAbeXP
17-963-404-3760
regular, even theodolites integrate carefully. bold,
special theodolites are slyly fluffily iron
7887.08                   Supplier#000009792
GERMANY
164759.00                  Manufacturer#3
Y28ITVeYriT3kIGdV2K8fSZ V2UqT5H1Otz
17-988-938-4296
pending, ironic packages sleep among the carefully
ironic accounts. quickly final accounts
7871.50                   Supplier#000007206
RUSSIA
104695.00                  Manufacturer#1
3w fNCnrVmvJjE95sgWZzvW
32-432-452-7731
furiously dogged pinto beans cajole. bold, express
notornis until the slyly pending
7852.45                   Supplier#000005864
RUSSIA
8363.00                   Manufacturer#4
WCNfBPZeSXh3h,c
32-454-883-3821
blithely regular deposits
7850.66                   Supplier#000001518
UNITED KINGDOM
86501.00                   Manufacturer#1
ONda3YJiHKJOC
33-730-383-3892
furiously final accounts wake carefully idle requests.
even dolphins wake acc
7843.52                   Supplier#000006683
FRANCE
11680.00                   Manufacturer#4
2Z0JGkiv01Y00oCFwUGfviIbhzCdy
16-464-517-8943
carefully bold accounts doub


100 rows processed.
Statement Processed in 4.63 seconds.

Ended Executing this Query at Tue Aug 19 12:31:58
2003


Query Started at 1061289114.09
Query Ended at 1061289118.72
Query Processed in 4.63 seconds


SQL statements processed: 1
Queries processed: 1

========================================
```

qual3.v1

```
========================================
Begin Execution at Tue Aug 19 12:31:58 2003


-- using default substitutions

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

```
where rownum <= 10

L_ORDERKEY                    REVENUE
O_ORDERDATE O_SHIPPRIORITY
2456423.00                  406181.01
05-MAR-95     0.00
3459808.00                  405838.70
04-MAR-95     0.00
492164.00                   390324.06
19-FEB-95     0.00
1188320.00                  384537.94
09-MAR-95     0.00
2435712.00                  378673.06
26-FEB-95     0.00
4878020.00                  378376.80
12-MAR-95     0.00
5521732.00                  375153.92
13-MAR-95     0.00
2628192.00                  373133.31
22-FEB-95     0.00
993600.00                   371407.46
05-MAR-95     0.00
2300070.00                  367371.15
13-MAR-95     0.00

10 rows processed.
Statement Processed in 0.99 seconds.

Ended Executing this Query at Tue Aug 19 12:31:59
2003

Query Started at 1061289118.87
Query Ended at 1061289119.86
Query Processed in 0.99 seconds


SQL statements processed: 1
Queries processed: 1

========================================
```

qual4.v1

```
========================================
Begin Execution at Tue Aug 19 12:32:00 2003

-- using default substitutions

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

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

5 rows processed.
Statement Processed in 0.90 seconds.

Ended Executing this Query at Tue Aug 19 12:32:00
2003

Query Started at 1061289120.01
Query Ended at 1061289120.91
Query Processed in 0.90 seconds

SQL statements processed: 1
Queries processed: 1

========================================
```

qual5.v1

```
========================================
Begin Execution at Tue Aug 19 12:32:01 2003

-- using default substitutions

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

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

5 rows processed.
Statement Processed in 4.43 seconds.

Ended Executing this Query at Tue Aug 19 12:32:05
2003

Query Started at 1061289121.06
Query Ended at 1061289125.49
Query Processed in 4.43 seconds

SQL statements processed: 1
```

```
Queries processed: 1

=========================================
```

qual6.v1

```
=========================================
Begin Execution at Tue Aug 19 12:32:05 2003


-- using default substitutions

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

REVENUE
123141078.23

1 row    processed.
Statement Processed in 0.28 seconds.

Ended Executing this Query at Tue Aug 19 12:32:05
2003

Query Started at 1061289125.65
Query Ended at 1061289125.94
Query Processed in 0.28 seconds

SQL statements processed: 1
Queries processed: 1

=========================================
```

qual7.v1

```
=========================================
Begin Execution at Tue Aug 19 12:32:06 2003

-- using default substitutions

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

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

4 rows processed.
Statement Processed in 5.82 seconds.

Ended Executing this Query at Tue Aug 19 12:32:11
2003

Query Started at 1061289126.09
Query Ended at 1061289131.91
Query Processed in 5.82 seconds

SQL statements processed: 1
Queries processed: 1

=========================================
```

qual8.v1

```
=========================================
Begin Execution at Tue Aug 19 12:32:12 2003

-- using default substitutions

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

O_YEAR                          MKT_SHARE
1995.00                       0.03
1996.00                       0.04

2 rows processed.
Statement Processed in 4.08 seconds.

Ended Executing this Query at Tue Aug 19 12:32:16
2003

Query Started at 1061289132.06
Query Ended at 1061289136.14
Query Processed in 4.08 seconds

SQL statements processed: 1
Queries processed: 1

========================================
```

qual9.v1

```
========================================
Begin Execution at Tue Aug 19 12:32:16 2003

-- using default substitutions

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

... rows truncated ...

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.
Statement Processed in 7.69 seconds.

Ended Executing this Query at Tue Aug 19 12:32:23
2003

Query Started at 1061289136.30
Query Ended at 1061289143.99
Query Processed in 7.69 seconds

SQL statements processed: 1
Queries processed: 1

========================================
```

qual10.v1

```
========================================
Begin Execution at Tue Aug 19 12:32:24 2003

-- using default substitutions

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

C_CUSTKEY                    C_NAME
REVENUE
C_ACCTBAL                    N_NAME
C_ADDRESS
C_PHONE
C_COMMENT
57040.00                  Customer#000057040
734235.25
632.87                    JAPAN
```

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

```
3869.25                      GERMANY
Az9RFaut7NkPnc5zSD2PwHgVwr4jRzq
17-945-916-9648
boldly final requests cajole blith
147946.00                    Customer#000147946
576455.13
2030.13                      ALGERIA
iANyZHjqhyy7Ajah0pTrYyhJ
10-886-956-3143
furiously even accounts are blithely above the furiousl
115640.00                    Customer#000115640
569341.19
6436.10                      ARGENTINA
Vtgfia9qI 7EpHgecU1X
11-411-543-4901
final instructions are slyly according to the
73606.00                     Customer#000073606
568656.86
1785.67                      JAPAN
xuR0Tro5yChDfOCrjkd2ol
22-437-653-6966
furiously bold orbits about the furiously busy requests
wake across the furiously quiet theodolites. d
110246.00                    Customer#000110246
566842.98
7763.35                      VIETNAM
7KzflgX MDOq7sOkI
31-943-426-9837
dolphins sleep blithely among the slyly final
142549.00                    Customer#000142549
563537.24
5085.99                      INDONESIA
ChqEoK43OysjdHbtKCp6dKqjNyvvi9
19-955-562-2398
regular, unusual dependencies boost slyly; ironic
attainments nag fluffily into the unusual packages?
146149.00                    Customer#000146149
557254.99
1791.55                      ROMANIA
s87fvzFQpU
29-744-164-6487
silent, unusual requests detect quickly slyly regul
52528.00                     Customer#000052528
556397.35
551.79                       ARGENTINA
NFztyTOR10UOJ
11-208-192-3205
unusual requests detect. slyly dogged theodolites use
slyly. deposit
23431.00                     Customer#000023431
554269.54
3381.86                      ROMANIA
HgiV0phqhaIa9aydNoIlb
29-915-458-2654
instructions nag quickly. furiously bold accounts cajol


20 rows processed.
Statement Processed in 2.87 seconds.

Ended Executing this Query at Tue Aug 19 12:32:27
2003

Query Started at 1061289144.14
Query Ended at 1061289147.01
Query Processed in 2.87 seconds

SQL statements processed: 1
Queries processed: 1

========================================
```

qual11.v1

```
========================================
Begin Execution at Tue Aug 19 12:32:27 2003

-- using default substitutions

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

PS_PARTKEY                VALUE
129760.00              17538456.86
166726.00              16503353.92
191287.00              16474801.97
161758.00              16101755.54
34452.00               15983844.72
139035.00              15907078.34
9403.00                15451755.62
154358.00              15212937.88
38823.00               15064802.86
85606.00               15053957.15
33354.00               14408297.40
154747.00              14407580.68
82865.00               14235489.78
76094.00               14094247.04
222.00                 13937777.74
121271.00              13908336.00
55221.00               13716120.47
22819.00               13666434.28
76281.00               13646853.68
85298.00               13581154.93
85158.00               13554904.00
139684.00              13535538.72
31034.00               13498025.25
87305.00               13482847.04
10181.00               13445148.75
62323.00               13411824.30
26489.00               13377256.38
96493.00               13339057.83
56548.00               13329014.97
55576.00               13306843.35
159751.00              13306614.48
92406.00               13287414.50
182636.00              13223726.74
199969.00              13135288.21
62865.00               13001926.94
7284.00                12945298.19
... rows truncated ...

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.
Statement Processed in 3.14 seconds.

Ended Executing this Query at Tue Aug 19 12:32:30 2003

Query Started at 1061289147.16
Query Ended at 1061289150.30
Query Processed in 3.14 seconds

SQL statements processed: 1
Queries processed: 1

========================================
```

qual12.v1

```
========================================
Begin Execution at Tue Aug 19 12:32:30 2003

-- using default substitutions

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

L_SHIPMODE HIGH_LINE_COUNT
LOW_LINE_COUNT
MAIL          6202.00                    9324.00
SHIP          6200.00                    9262.00

2 rows processed.
Statement Processed in 0.79 seconds.

Ended Executing this Query at Tue Aug 19 12:32:31
2003

Query Started at 1061289150.45
Query Ended at 1061289151.24
Query Processed in 0.79 seconds

SQL statements processed: 1
Queries processed: 1
```

===

qual13.v1

===

```
Begin Execution at Tue Aug 19 12:32:31 2003

-- using default substitutions

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

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

42 rows processed.
Statement Processed in 1.86 seconds.

Ended Executing this Query at Tue Aug 19 12:32:33
2003

Query Started at 1061289151.40
Query Ended at 1061289153.26
Query Processed in 1.86 seconds


SQL statements processed: 1
Queries processed: 1
```

===

qual14.v1

===

```
Begin Execution at Tue Aug 19 12:32:33 2003

-- using default substitutions

select
          100.00 * sum(case
                    when p_type like 'PROMO%'
                              then l_extendedprice *
(1 - l_discount)
                    else 0
```

```
        end) / sum(l_extendedprice * (1 - l_discount))
as promo_revenue
from
        lineitem,
        part
where
        l_partkey = p_partkey
        and l_shipdate >= date '1995-09-01'
        and l_shipdate < date '1995-09-01' + interval '1'
month

PROMO_REVENUE
16.38

1 row   processed.
Statement Processed in 0.38 seconds.

Ended Executing this Query at Tue Aug 19 12:32:33
2003

Query Started at 1061289153.41
Query Ended at 1061289153.79
Query Processed in 0.38 seconds

SQL statements processed: 1
Queries processed: 1

========================================
```

qual15.v1

```
========================================
Begin Execution at Tue Aug 19 12:32:33 2003

-- using default substitutions

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

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

1 row   processed.
Statement Processed in 10.39 seconds.

Ended Executing this Query at Tue Aug 19 12:32:44
2003

Query Started at 1061289153.94
Query Ended at 1061289164.33
Query Processed in 10.39 seconds

SQL statements processed: 1
Queries processed: 1

========================================
```

qual16.v1

```
========================================
Begin Execution at Tue Aug 19 12:32:44 2003

-- using default substitutions

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

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

```
Brand#15     SMALL BURNISHED NICKEL        19.00
24.00
Brand#21     MEDIUM ANODIZED COPPER         3.00
24.00
Brand#22     SMALL BRUSHED NICKEL          3.00
24.00
Brand#22     SMALL BURNISHED BRASS         19.00
24.00
Brand#25     MEDIUM BURNISHED COPPER
36.00                    24.00
Brand#31     PROMO POLISHED COPPER
36.00                    24.00
Brand#33     LARGE POLISHED TIN            23.00
24.00
Brand#33     PROMO POLISHED STEEL          14.00
24.00
Brand#35     PROMO BRUSHED NICKEL          14.00
24.00
Brand#41     ECONOMY BRUSHED STEEL          9.00
24.00
Brand#41     ECONOMY POLISHED TIN          19.00
24.00
Brand#41     LARGE PLATED COPPER           36.00
24.00
Brand#42     ECONOMY PLATED BRASS           3.00
24.00
Brand#42     STANDARD POLISHED TIN        49.00
24.00
Brand#43     PROMO BRUSHED TIN             3.00
24.00
Brand#43     SMALL ANODIZED COPPER         36.00
24.00
Brand#44     STANDARD POLISHED NICKEL   3.00
24.00
Brand#52     ECONOMY PLATED TIN           14.00
24.00
Brand#52     STANDARD BURNISHED NICKEL 3.00
24.00
Brand#53     MEDIUM ANODIZED STEEL         14.00
24.00
Brand#14     PROMO ANODIZED NICKEL         45.00
23.00
Brand#32     ECONOMY PLATED BRASS           9.00
23.00
Brand#52     SMALL ANODIZED COPPER         3.00
23.00
Brand#11     ECONOMY BRUSHED COPPER
45.00                    20.00
Brand#11     ECONOMY PLATED BRASS          23.00
20.00
Brand#11     LARGE BRUSHED COPPER          49.00
20.00
Brand#11     LARGE POLISHED COPPER         49.00
20.00

... rows truncated ...

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.
Statement Processed in 3.11 seconds.

Ended Executing this Query at Tue Aug 19 12:32:47
2003

Query Started at 1061289164.48
Query Ended at 1061289167.59
Query Processed in 3.11 seconds

SQL statements processed: 1
Queries processed: 1

========================================
```

qual17.v1

```
========================================
Begin Execution at Tue Aug 19 12:32:47 2003

-- using default substitutions

select
sum(l_extendedprice) / 7.0 as avg_yearly
from
lineitem,
part
where
p_partkey = l_partkey
and p_brand = 'Brand#23'
and p_container = 'MED BOX'
and l_quantity < (
select
0.2 * avg(l_quantity)
from
lineitem
where
l_partkey = p_partkey
)
AVG_YEARLY
348406.05

1 row    processed.
Statement Processed in 3.10 seconds.

Ended Executing this Query at Tue Aug 19 12:32:50
2003

Query Started at 1061289167.76
Query Ended at 1061289170.86
Query Processed in 3.10 seconds

SQL statements processed: 1
Queries processed: 1

========================================
```

qual18.v1

```
========================================
Begin Execution at Tue Aug 19 12:32:51 2003

-- using default substitutions

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

C_NAME                          C_CUSTKEY
O_ORDERKEY                    O_ORDERDATE
O_TOTALPRICE                 SUM(L_QUANTITY)
Customer#000128120        128120.00
4722021.00                07-APR-94
544089.09                 323.00
Customer#000144617        144617.00
3043270.00                12-FEB-97
530604.44                 317.00
Customer#000013940         13940.00
2232932.00                13-APR-97
522720.61                 304.00
Customer#000066790         66790.00
2199712.00                30-SEP-96
515531.82                 327.00
Customer#000046435         46435.00
4745607.00                03-JUL-97
508047.99                 309.00
Customer#000015272         15272.00
3883783.00                28-JUL-93
500241.33                 302.00
Customer#000146608        146608.00
3342468.00                12-JUN-94
499794.58                 303.00
Customer#000096103         96103.00
5984582.00                16-MAR-92
494398.79                 312.00
Customer#000024341         24341.00
1474818.00                15-NOV-92
491348.26                 302.00
Customer#000137446        137446.00
5489475.00                23-MAY-97
487763.25                 311.00
Customer#000107590        107590.00
4267751.00                04-NOV-94
485141.38                 301.00
Customer#000050008         50008.00
2366755.00                09-DEC-96
483891.26                 302.00
Customer#000015619         15619.00
3767271.00                07-AUG-96
480083.96                 318.00
Customer#000077260         77260.00
1436544.00                12-SEP-92

... rows truncated ...

436275.31                 305.00
Customer#000081581         81581.00
4739650.00                04-NOV-95
435405.90                 305.00
Customer#000119989        119989.00
1544643.00                20-SEP-97
434568.25                 320.00
Customer#000003680          3680.00
3861123.00                03-JUL-98
433525.97                 301.00
Customer#000113131        113131.00
967334.00                 15-DEC-95
432957.75                 301.00
```

```
Customer#000141098          141098.00
565574.00               24-SEP-95
430986.69               301.00
Customer#000093392          93392.00
5200102.00              22-JAN-97
425487.51               304.00
Customer#000015631          15631.00
1845057.00              12-MAY-94
419879.59               302.00
Customer#000112987          112987.00
4439686.00              17-SEP-96
418161.49               305.00
Customer#000012599          12599.00
4259524.00              12-FEB-98
415200.61               304.00
Customer#000105410          105410.00
4478371.00              05-MAR-96
412754.51               302.00
Customer#000149842          149842.00
5156581.00              30-MAY-94
411329.35               302.00
Customer#000010129          10129.00
5849444.00              21-MAR-94
409129.85               309.00
Customer#000069904          69904.00
1742403.00              19-OCT-96
408513.00               305.00
Customer#000017746          17746.00
6882.00                 09-APR-97
408446.93               303.00
Customer#000013072          13072.00
1481925.00              15-MAR-98
399195.47               301.00
Customer#000082441          82441.00
857959.00               07-FEB-94
382579.74               305.00
Customer#000088703          88703.00
2995076.00              30-JAN-94
363812.12               302.00

57 rows processed.
Statement Processed in 2.02 seconds.

Ended Executing this Query at Tue Aug 19 12:32:53
2003

Query Started at 1061289171.01
Query Ended at 1061289173.03
Query Processed in 2.02 seconds

SQL statements processed: 1
Queries processed: 1

========================================
```

qual19.v1

```
========================================
Begin Execution at Tue Aug 19 12:32:53 2003

-- using default substitutions

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

REVENUE
3083843.06

1 row    processed.
Statement Processed in 2.06 seconds.

Ended Executing this Query at Tue Aug 19 12:32:55
2003

Query Started at 1061289173.18
Query Ended at 1061289175.24
Query Processed in 2.06 seconds

SQL statements processed: 1
Queries processed: 1

========================================
```

qual20.v1

```
========================================
Begin Execution at Tue Aug 19 12:32:55 2003

-- using default substitutions

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

S_NAME                          S_ADDRESS
Supplier#000000020
iybAE,RmTymrZVYaFZva2SH,j
Supplier#000000091
YV45D7TkfdQanOOZ7q9QxkyGUapU1oOWU6q3
Supplier#000000197
YC2Acon6kjY3zj3Fbxs2k4Vdf7X0cd2F
Supplier#000000226
83qOdU2EYRdPQAQhEtn GRZEd
Supplier#000000285
Br7e1nnt1yxrw6ImgpJ7YdhFDjuBf
Supplier#000000378          FfbhyCxWvcPrO8ltp9
Supplier#000000402
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

... rows truncated ...

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.
Statement Processed in 3.96 seconds.

Ended Executing this Query at Tue Aug 19 12:32:59
2003

Query Started at 1061289175.39
Query Ended at 1061289179.35
Query Processed in 3.96 seconds

SQL statements processed: 1
Queries processed: 1

========================================
```

qual21.v1

```
========================================
Begin Execution at Tue Aug 19 12:32:59 2003

-- using default substitutions

select * from (
select
s_name,
count(*)    numwait
from
supplier,
lineitem l1,
orders,
nation
where
s_suppkey = l1.l_suppkey
and o_orderkey = l1.l_orderkey
and o_orderstatus = 'F'
and l1.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
... rows truncated ...

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.
Statement Processed in 5.58 seconds.

Ended Executing this Query at Tue Aug 19 12:33:05
2003

Query Started at 1061289179.50
Query Ended at 1061289185.08
Query Processed in 5.58 seconds

SQL statements processed: 1
Queries processed: 1

========================================
```

qual22.v1

```
========================================
Begin Execution at Tue Aug 19 12:33:05 2003

-- using default substitutions

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

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

7 rows processed.
Statement Processed in 0.99 seconds.

Ended Executing this Query at Tue Aug 19 12:33:06
2003

Query Started at 1061289185.23
Query Ended at 1061289186.23
Query Processed in 0.99 seconds


SQL statements processed: 1
Queries processed: 1
```

Appendix D. Seed and Query Substitution Parameters

This Appendix contains Seed values and substitution parameters for each stream.

```
This Appendix contains Seed values and
substitution parameters for each stream

======================================
seed values
======================================
session 00 905102702
session 01 905102703
session 02 905102704
session 03 905102705
session 04 905102706
session 05 905102707
session 06 905102708
session 07 905102709
======================================
stream 00 substitution parameters
======================================
14       1994-09-01
2        39       NICKEL   AFRICA
9        pale
20       maroon   1996-01-01        VIETNAM
6        1994-01-01        0.06     24
17       Brand#13 JUMBO DRUM
18       313
8        INDONESIA         ASIA     LARGE
BRUSHED STEEL
21       MOROCCO
13       unusual  requests
3        HOUSEHOLD         1995-03-04
22       34       23       14       18
         24       26       32
16       Brand#51 MEDIUM BRUSHED    5
         28       2        45       21
         8        18       48
4        1993-08-01
11       SAUDI ARABIA      0.0000001000
15       1994-03-01
1        88
10       1994-03-01
19       Brand#43 Brand#45 Brand#35 4
         11       27
5        AFRICA   1994-01-01
7        JORDAN   INDONESIA
12       REG AIR  MAIL     1994-01-01
======================================
stream 01 substitution parameters
======================================
21       GERMANY
3        AUTOMOBILE        1995-03-21
18       315
5        AMERICA  1994-01-01
11       INDIA    0.0000001000
7        ETHIOPIA ARGENTINA
6        1994-01-01        0.03     24
20       turquoise         1994-01-01
         IRAQ
17       Brand#15 WRAP BAG
12       SHIP     MAIL     1994-01-01
16       Brand#41 PROMO BURNISHED   43
         48       30       26       24
         20       27       42
15       1996-10-01
13       unusual  requests
10       1994-12-01
2        26       TIN      EUROPE
8        ARGENTINA         AMERICA  LARGE
PLATED STEEL
14       1995-01-01
19       Brand#55 Brand#23 Brand#24 9
         12       23
9        moccasin
22       21       12       24       19
         15       17       14
1        96
4        1996-03-01
======================================
stream 02 substitution parameters
======================================
6        1994-01-01        0.09     25
17       Brand#12 WRAP PACK
14       1995-04-01
16       Brand#21 SMALL PLATED      46
         19       42       50       14
         16       21       48
19       Brand#52 Brand#51 Brand#24 5
         13       30
10       1993-09-01
9        maroon
2        14       COPPER   AFRICA
15       1994-07-01
8        CHINA    ASIA     LARGE ANODIZED
STEEL
5        EUROPE   1994-01-01
22       11       14       22       20
         33       13       28
12       MAIL     FOB      1995-01-01
7        RUSSIA   CHINA
13       unusual  requests
18       313
1        104
4        1993-12-01
20       green    1997-01-01
         ARGENTINA
3        FURNITURE         1995-03-06
11       VIETNAM  0.0000001000
21       UNITED STATES
======================================
stream 03 substitution parameters
======================================
8        IRAN     MIDDLE EAST       MEDIUM
POLISHED COPPER
5        MIDDLE EAST       1994-01-01
4        1996-07-01
6        1994-01-01        0.06     24
17       Brand#14 WRAP DRUM
```

```
7       KENYA   IRAN
1       112
18      314
22      25      29      27      33
        16      10      11
14      1995-07-01
9       lawn
10      1994-07-01
15      1997-01-01
11      INDONESIA       0.0000001000
20      rosy    1996-01-01      MOROCCO
2       2       BRASS   EUROPE
21      MOZAMBIQUE
19      Brand#54 Brand#44 Brand#23 10
        14      26
13      express requests
16      Brand#51 LARGE BRUSHED  50
        45      34      6       18
        33      47      20
12      TRUCK   FOB     1995-01-01
3       AUTOMOBILE      1995-03-23
=====================================
```

stream 04 substitution parameters

```
=====================================
5       AFRICA  1995-01-01
21      INDONESIA
14      1995-10-01
19      Brand#11 Brand#22 Brand#12 5
        15      23
15      1994-10-01
17      Brand#11 SM BAG
12      RAIL    FOB     1995-01-01
6       1995-01-01      0.03    24
4       1994-04-01
9       hot
8       BRAZIL  AMERICA MEDIUM BURNISHED
COPPER
16      Brand#41 STANDARD ANODIZED 14
        10      1       12      16
        9       24      25
11      RUSSIA  0.0000001000
2       40      NICKEL  AMERICA
10      1993-04-01
18      312
1       120
13      express accounts
7       FRANCE  BRAZIL
22      32      13      18      21
        22      19      10
3       FURNITURE       1995-03-08
20      cornsilk 1994-01-01     ETHIOPIA
=====================================
```

stream 05 substitution parameters

```
=====================================
21      ARGENTINA
15      1997-05-01
4       1996-11-01
6       1995-01-01      0.09    25
7       UNITED KINGDOM  ROMANIA
16      Brand#21 MEDIUM PLATED  6
        40      3       5       19
        9       50      30
19      Brand#14 Brand#15 Brand#11 10
        16      30
18      313
14      1996-01-01
22      20      11      28      17
        29      33      13
11      IRAN    0.0000001000
13      express accounts
3       MACHINERY       1995-03-25
1       67
2       28      TIN     EUROPE
5       AMERICA 1995-01-01
8       ROMANIA EUROPE  SMALL BRUSHED
COPPER
20      navy    1993-01-01      SAUDI
ARABIA
12      AIR     FOB     1995-01-01
17      Brand#13 SM PACK
10      1994-01-01
9       gainsboro
=====================================
```

stream 06 substitution parameters

```
=====================================
10      1994-10-01
3       FURNITURE       1995-03-10
15      1995-02-01
13      express accounts
6       1995-01-01      0.06    24
8       IRAQ    MIDDLE EAST     SMALL
PLATED COPPER
9       dodger
7       MOROCCO IRAQ
4       1994-08-01
11      UNITED KINGDOM  0.0000001000
22      12      13      11      22
        25      27      20
18      315
12      REG AIR SHIP    1996-01-01
1       75
5       ASIA    1995-01-01
16      Brand#11 ECONOMY POLISHED 6
        2       31      20      10
        5       32      34
2       15      COPPER  AMERICA
14      1996-05-01
19      Brand#11 Brand#42 Brand#11 6
        17      26
20      azure   1996-01-01
        INDONESIA
17      Brand#14 SM DRUM
21      CHINA
=====================================
```

stream 07 substitution parameters

```
=====================================
18      312
8       CANADA  AMERICA SMALL ANODIZED
COPPER
20      lavender 1995-01-01     UNITED
STATES
21      IRAN
2       3       STEEL   MIDDLE EAST
4       1997-03-01
22      34      13      12      33
        28      16      20
17      Brand#11 LG BAG
1       83
11      IRAQ    0.0000001000
9       cornsilk
19      Brand#23 Brand#35 Brand#55 1
        18      22
3       MACHINERY       1995-03-27
13      express accounts
5       EUROPE  1995-01-01
7       GERMANY CANADA
10      1993-07-01
16      Brand#41 SMALL ANODIZED 36
        23      34      50      25
        44      4       38
6       1995-01-01      0.04    24
14      1996-08-01
15      1997-08-01
12      FOB     SHIP    1996-01-01
```

Appendix E. Implementation-Specific Layer/Driver Code

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

ECHO=echo

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

RUN_ID_FILE=${KIT_DIR}/audit/r_id

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

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

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

SCRIPT_LOG_FILE=${OUT_DIR}/main.out
RDB_TABLES=${OUT_DIR}/rdbtablest
FIRST_TEN=${OUT_DIR}/firstten
LD1DBCRE=${OUT_DIR}/Ld1dbcre
LD2SCTSO=${OUT_DIR}/Ld2sctso
LD3DAPOP=${OUT_DIR}/Ld4dapop
LD4IXCRE=${OUT_DIR}/Ld5ixcre
LD5ANLYZ=${OUT_DIR}/Ld5anlyz
#DAT_FILE=${TPCH}/bmc/schema/3tb/var9/3tb_1
6.dat
#DAT_FILE=${TPCH}/bmc/schema/3tb/var9/3tb_1
28.dat
DAT_FILE=${TPCH}/bmc/schema/1tb/var9/1tb.da
t

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

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

echo "Start: load database `date`" >>
$SCRIPT_LOG_FILE
bumpx.pl -s -x -o ${DAT_FILE} -p dbcre >
$LD1DBCRE
bumpx.pl -s -x -o ${DAT_FILE} -p sctso >
$LD2SCTSO
STIME=`$GTIME`
echo "Start: timed load portion `date`" >>
$SCRIPT_LOG_FILE
bumpx.pl -s -x -o ${DAT_FILE} -p dapop >
$LD3DAPOP
bumpx.pl -s -x -o ${DAT_FILE} -p ixcre >
$LD4IXCRE
bumpx.pl -s -x -o ${DAT_FILE} -p anlyz >
$LD5ANLYZ
# tshut
# tstart
ckpnt.sh
echo "End: timed load portion `date`" >>
$SCRIPT_LOG_FILE

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

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

$BMC/scripts/set_dop.sh 128

tshut >>  $SCRIPT_LOG_FILE
tstart >>  $SCRIPT_LOG_FILE
# ckpnt.sh
```

```
===========================================
runTPCHall_run1
===========================================
#!/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


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=${OUT_DIR}/main.out
RDB_TABLES=${OUT_DIR}/rdbtablest
FIRST_TEN=${OUT_DIR}/firstten
```

```
LD1DBCRE=${OUT_DIR}/Ld1dbcre
LD2SCTSO=${OUT_DIR}/Ld2sctso
LD3DAPOP=${OUT_DIR}/Ld4dapop
LD4IXCRE=${OUT_DIR}/Ld5ixcre
LD5ANLYZ=${OUT_DIR}/Ld5anlyz
DAT_FILE=${TPCH}/bmc/schema/100g_84.dat


runTPCHpt ${SCALE_FACTOR} 1 ${RUN_ID}

#tshut >>  $SCRIPT_LOG_FILE
#tstart >>  $SCRIPT_LOG_FILE
ckpnt.sh
```

===

runTPCHpt

===

```
#!/bin/ksh
. $KIT_DIR/env
#set -x
#ECHOS=/bin/echo
SCRIPT_DIR=${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_STR
EAMS-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}tstrcn
t

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}

##lm
$ECHOS mystartstat ${RUN_ID} pwr_${PARA}

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

${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
${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

##lm
$ECHOS mystopstat ${RUN_ID} pwr_${PARA}

${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${PA
RA}.${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

##lm
$ECHOS mystartstat ${RUN_ID} thr_${PARA}

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

##lm
$ECHOS mystopstat ${RUN_ID} thr_${PARA}

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}.rp
t
```

===

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
```

===

runuf1.sh

===

```
#!/bin/ksh
. $KIT_DIR/env
O=${ORACLE_HOME}
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
UTILS_DIR=${KIT_DIR}/utils
LOG_DIR=${UPDATE_DIR}/log
GTIME=${UTILS_DIR}/gtime
SF=${SCALE_FACTOR}
PAR_HINT=${UPDATE_DOP_INS} # is defined in env
file

LOGPATH=.
PASSWD=${DATABASE_USER}

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


START=`$GTIME`


sqlplus /NOLOG << !

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

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

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

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

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

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

insert into lineitem (
  select
    l_shipdate         ,
    l_orderkey         ,
    l_discount         ,
    l_extendedprice     ,
    l_suppkey          ,
    l_quantity         ,
    l_returnflag       ,
    l_partkey          ,
    l_linestatus       ,
    l_tax             ,
    l_commitdate       ,
    l_receiptdate      ,
    l_shipmode         ,
    l_linenumber       ,
    l_shipinstruct      ,
    l_comment
from temp_l_et);

commit;

drop table temp_l_et;
drop table temp_o_et;

exit;
!

END=`$GTIME`


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

==
=====

runuf2.sh

==

```
#!/bin/ksh
. $KIT_DIR/env
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
UTILS_DIR=${KIT_DIR}/utils
GTIME=${UTILS_DIR}/gtime
LOG_DIR=${UPDATE_DIR}/log
PAR_HINT=${UPDATE_DOP_DEL} #defined in env
file
SF=${SCALE_FACTOR}
PASSWD=${DATABASE_USER}

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

SETNUM=$1

i=1
PID=""

START=`$GTIME`

sqlplus /NOLOG << !

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

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

drop table temp_okey_et;
drop table temp_okey;

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

--alter table temp_okey_et parallel
${PAR_HINT};
alter table temp_okey_et parallel 16;

--create table temp_okey parallel ${PAR_HINT}
nologging as select * from temp_okey_et;
create table temp_okey parallel 16 nologging
as select * from temp_okey_et;

--create unique index i_temp_okey on temp_okey
(t_orderkey) parallel ${PAR_HINT} nologging
compute statistics;
create unique index i_temp_okey on temp_okey
(t_orderkey) parallel 16 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=10;

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

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

commit;

drop table temp_okey;
drop table temp_okey_et;
exit;
!

END=`$GTIME`


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

```
===========================================
```

qexecpl.c

```
===========================================
#ifdef RCSID
static char *RCSid =
#endif /* RCSID */

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

/*

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

  DESCRIPTION
    SQL Execution Engine, Oracle v8,  OCI
version

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

  MODIFIED   (MM/DD/YY)

*/

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

#include "qexecpl.h"

/* Function Prototypes */

extern double gettime();

/* function prototypes from gen.c */

int get_statement();

/* Declare error handling functions */

void sql_error();

/* Other prototypes */

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

char logname[UNAME_LEN];  /* username/passwd
combo */
char *passwd;
double tr_start = 0.0;    /* query start time
*/
double tr_end = 0.0;      /* query end time
*/

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

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

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

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

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

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

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

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

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

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

/* OCI handles */

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

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

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

void usage() {

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



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

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

  switch(status) {
  case OCI_SUCCESS_WITH_INFO:
    fprintf(stderr, "Error: Statement
returned with info.\n");
    if (type)
      (void)
OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(te
xt*)msg,

2048,OCI_HTYPE_ERROR);
    else
      (void)
OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(te
xt*)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,(te
xt*)msg,

2048,OCI_HTYPE_ERROR);
    else
      (void)
OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(te
xt*)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,(te
xt*)msg,

2048,OCI_HTYPE_ERROR);
    else
      (void)
OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(te
xt*)msg,

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

  /* Rollback just in case */

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

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

  SQLexit();

  exit(1);
}

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

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

```
#endif
  /* Initialize some variables */

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

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

  strcpy(logname, argv[1]);

  /* Process  optional parameters */

  argc -= 1;
  argv += 1;

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

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

  SQLinit();

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

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

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

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

    switch (retcode) {

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

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

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

      /* Executes the query */

      SQLexec();

      s_tr_end = gettime();
      stmt_cnt++;

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

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

  /* Get Timing for the last query */

  tr_end = gettime();

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

  fprintf(rep, "%.2f\n", (tr_end - tr_start));
  fprintf(logfile, "\nSQL statements
processed: %d\n", stmt_cnt);
  fprintf(logfile, "Queries processed: %d\n",
qry_cnt);

  fflush(rep);
  fflush(logfile);

  /* Close the query template file */

  fclose(qtemp);

  /*  Disconnect from ORACLE. */

  SQLexit();
  exit(0);
}



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

void SQLinit() {

  int i;

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

  for (i=0; i<MAX_SEL_LIST; i++) {
    if (i < MAX_PREALLOC) {
      dlist[i] = (dltype *) memalloc
(sizeof(dltype));
      dlist[i]->defhdl = NULL;
/*
OCIhalloc(curq,&(dlist[i]->defhdl),OCI_HTYP
E_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,OC
I_ATTR_SERVER,errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,logname,st
rlen(logname),OCI_ATTR_USERNAME,
          errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,str
len(passwd),OCI_ATTR_PASSWORD,
          errhp);

  if ((status = OCISessionBegin(tpcsvc, errhp,
tpcusr, OCI_CRED_RDBMS,

         OCI_DEFAULT)) != OCI_SUCCESS)
    sql_error(errhp,status,1);


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcusr,0,OC
I_ATTR_SESSION,errhp);

  /* Enable session parallel dml */

  sprintf((char *) stmt, PDMLTXT);

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

  /* Enable session parallel ddl */

  sprintf((char *) stmt, PDDLTXT);

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

  /* set serializable level */

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

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

}




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

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

  if (!end_flag) {

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

    tr_end = gettime();

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


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

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

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

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

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

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

      fflush(logfile);
```

```
      fflush(rep);

    }

    tr_start = tr_end;
    qry_cnt++;

    end_flag = 1;
  }

  s_tr_start = gettime();

  /* prepare the statement */

  if ((status = OCIStmtPrepare(curq, errhp,
(text*) stmt, (ub4) strlen(stmt),

OCI_NTV_SYNTAX, OCI_DEFAULT)) != OCI_SUCCESS)
    sql_error(errhp,status,1);

  /* Prints the query text to the logfile */

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

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


OCIaget(curq,OCI_HTYPE_STMT,&stmttyp,NULL,O
CI_ATTR_STMT_TYPE,errhp);

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

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

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

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

  OCIsexec(tpcsvc,curq,errhp,0);

  num_sel_list = define_output_variables();

  /* Executes the query and fetches the rows
*/

  (void) process_select_list(num_sel_list);

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


OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,
OCI_ATTR_ROW_COUNT,errhp);

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

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

  /* reset set_fetchrows */

  num_to_fetch = -1;

}


void SQLexit() {

  int i;

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

  /* free all memory */

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

  /* Flush all output */

  fflush(rep);
  fflush(logfile);

}


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

int define_output_variables()
{

  int i;
  int retflag = 0;

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

    slist[i].buflen = MAX_COLNAME_SIZE;

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


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

    OCIaget(tpcpar, OCI_DTYPE_PARAM,
&(slist[i].dbsize),
            NULL, OCI_ATTR_DATA_SIZE,
errhp);
```

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

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

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

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

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

   switch (slist[i].dbtype) {

   case OCI_TYPECODE_NUMBER:

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

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

     break;

   default:

     /* default is character string */

     defbuf = (char **) dlist[i]->sbuf;
```

```
      deflen = MAX_STR_LEN;
      deftype = SQLT_STR;
/*     deftype = OCI_TYPECODE_CHAR; */
      break;
    }

    /* Define the column */

    if
((status=OCIDefineByPos(curq,&(dlist[i]->de
fhdl),errhp,POS(i),

defbuf,deflen,deftype,NULL,

dlist[i]->rlen,NULL,OCI_DEFAULT))!=OCI_SUCC
ESS)
      sql_error(errhp,status,1);
  }
  return i;
}


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

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

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

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

  print_header(num);

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

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

  /* Fetch the rows and print them out */

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

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


OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,
OCI_ATTR_ROW_COUNT,errhp);

    print_rows(num,rows_ret);

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

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

OCI_DEFAULT)) == OCI_SUCCESS) {

OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NUL
L,

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_s
o_far,NULL,

OCI_ATTR_ROW_COUNT,errhp);
           print_rows(num,(num_so_far-rows_re
t));
           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,NUL
L,

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_s
o_far,NULL,

OCI_ATTR_ROW_COUNT,errhp);
           print_rows(num,(num_so_far-rows_re
t));
           rows_ret = num_so_far;
      }
    }
  } else {
    OCIStmtFetch(curq, errhp, ntf,
OCI_FETCH_NEXT, OCI_DEFAULT);

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

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

}


int get_statement()
{

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

  /* Reset statement buffer */

  stmt[0] = '\0';
  while (fgets(line, 127, qtemp) != NULL) {

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

    /* remove blanks */

    str = line;

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

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

    strcat(stmt, str);

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

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

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


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

void *memalloc(size)
     int size;
{

  void *tmp;

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



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

  int i, diff;
  char colname[MAX_COLNAME_SIZE];
```

```
  int len = 0;          /* Running column length
*/
  int cwid = 0;

  fprintf(logfile, "\n");

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

    /* extract the column name */

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

    /* format the output a little */

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

    /* do a little bit of formatting */

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

  fprintf(logfile,"\n");
}


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

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

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

    len = 0;

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

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

      /* do a little bit of formatting */

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

```
         len = cwid;
     }

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



void remove_newline(str)
    char *str;
{

  char *p;

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



===========================================
```

qexecpl.h

```
===========================================
/*
 */

/* 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)

*/

/*
# 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,it
er,0,NULL,NULL,OCI_DEFAULT)) != OCI_SUCCESS)
\
      sql_error(errh,status,1); \
   else \
      DISCARD 0


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

#endif /* QSTREAMPL_H */
```

```
===========================================
```

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)

*/
#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 */
```

Appendix F. Misc database scripts

Activity between Database Load and Run1. When the load finished, the runTPCHall script automatically selected a seed value and saved it.

The database was restarted.

Then the 2 auditor scripts count.sql and dbtables.sql were run to validate that the database structure was correct.

```
========================================
```

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;
```

```
========================================
```

dbtables.sql

```
========================================
set echo on
set numwidth 25
spool rdbtablest
SELECT COUNT(*) FROM LINEITEM;

SELECT * FROM LINEITEM
WHERE L_ORDERKEY IN
  ( 4, 26598,    148577,    387431, 56704, 517442,
600000)
AND L_LINENUMBER = 1
ORDER BY L_ORDERKEY;



SELECT * FROM REGION;

SELECT COUNT(*) FROM NATION;

SELECT * FROM NATION
WHERE N_NATIONKEY IN (3,10,14,20)
ORDER BY N_NATIONKEY;



SELECT COUNT(*) FROM ORDERS;

SELECT * FROM ORDERS
WHERE O_ORDERKEY IN ( 7, 44065, 287590,
411111, 483876, 599942 )
ORDER BY O_ORDERKEY;



SELECT COUNT(*) FROM PART;

SELECT * FROM PART
WHERE P_PARTKEY IN
(1,984,8743,9028,13876,17899,20000)
ORDER BY P_PARTKEY;



SELECT COUNT(*) FROM PARTSUPP;

SELECT* FROM PARTSUPP
    WHERE PS_PARTKEY = 3398
    AND PS_SUPPKEY = (SELECT
MIN(PS_SUPPKEY)
        FROM PARTSUPP WHERE PS_PARTKEY =
3398);

SELECT* FROM PARTSUPP
    WHERE PS_PARTKEY =15873
    AND PS_SUPPKEY = (SELECT
MIN(PS_SUPPKEY)
        FROM PARTSUPP WHERE PS_PARTKEY =
15873);

SELECT* FROM PARTSUPP
    WHERE PS_PARTKEY = 11394
    AND PS_SUPPKEY = (SELECT
MIN(PS_SUPPKEY)
        FROM PARTSUPP WHERE PS_PARTKEY =
11394);

SELECT* FROM PARTSUPP
    WHERE PS_PARTKEY = 6743
    AND PS_SUPPKEY = (SELECT
MIN(PS_SUPPKEY)
        FROM PARTSUPP WHERE PS_PARTKEY =
6743);

SELECT* FROM PARTSUPP
    WHERE PS_PARTKEY = 19763
    AND PS_SUPPKEY = (SELECT
MIN(PS_SUPPKEY)
```

```
        FROM PARTSUPP WHERE PS_PARTKEY =19763);



SELECT COUNT(*) FROM SUPPLIER;

SELECT * FROM SUPPLIER
WHERE   S_SUPPKEY IN (83,265,492,784,901,1000)
ORDER BY S_SUPPKEY;

DROP TABLE MINMAX;

CREATE TABLE MINMAX
(TNAME CHAR(15),
KEYMIN INTEGER,
KEYMAX INTEGER);

INSERT INTO MINMAX
SELECT
'LINEITEM_ORD',MIN(L_ORDERKEY),MAX(L_ORDERKEY)
FROM LINEITEM ;

INSERT INTO MINMAX
SELECT
'LINEITEM_NBR',MIN(L_LINENUMBER),MAX(L_LINENUMBER)
FROM LINEITEM;

INSERT INTO MINMAX
SELECT
'ORDERTBL',MIN(O_ORDERKEY),MAX(O_ORDERKEY)
FROM ORDERS;

INSERT INTO MINMAX
SELECT
'CUSTOMER',MIN(C_CUSTKEY),MAX(C_CUSTKEY)
FROM CUSTOMER;

INSERT INTO MINMAX
SELECT
'PART',MIN(P_PARTKEY),MAX(P_PARTKEY)
FROM PART;

INSERT INTO MINMAX
SELECT
'SUPPLIER',MIN(S_SUPPKEY),MAX(S_SUPPKEY)
FROM SUPPLIER;

INSERT INTO MINMAX
SELECT
'PARTSUPP_PART',MIN(PS_PARTKEY),MAX(PS_PARTKEY)
FROM PARTSUPP;

INSERT INTO MINMAX
SELECT
'PARTSUPP_SUPP',MIN(PS_SUPPKEY),MAX(PS_SUPPKEY)
FROM PARTSUPP ;

INSERT INTO MINMAX
SELECT
'NATION',MIN(N_NATIONKEY),MAX(N_NATIONKEY)
FROM NATION;

INSERT INTO MINMAX
SELECT
'REGION',MIN(R_REGIONKEY),MAX(R_REGIONKEY)
FROM REGION;

SELECT * FROM MINMAX;
spool off
exit;


========================================
```

tshut

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

if [ "$2" != "" -a "$2" != "1" ]; then
   INUM=$2
   if [ -f $ORACLE_HOME/work/t_init$INUM.ora ]; then
      export ORACLE_SID="$ORACLE_SID"$INUM
   fi
fi

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

========================================
```

tstart

```
========================================
#!/bin/ksh
#
# $Header: ostart.sh 17-aug-99.16:20:43 mpoess Exp $
# ostart.sh
#   Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#     NAME
#        ostart.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        08/17/99 - Creation
#     mpoess        08/17/99 - Creation
#

cd $TPCH/admin

if [ "$1" = "-p" ]; then
. ./pgenv
fi

echo "booting oracle"
pwd
#exit
```

```
sqlplus /nolog << !
connect / as sysdba
shutdown immediate
startup pfile=init.ora
exit
!
```

Appendix G. Pricing Information

For Veritas pricing please contact:

Michael Nawson
+1-(1) 407-357-5193
mike.mawson@veritas.com

For Oracle pricing please contact:

MaryBeth Pierantoni
+1- (1)650-506-2118
mary.beth.pierantoni@oracle.com

For Fujitsu-Siemens pricing please contact:

Jürgen Binder
+49-(0)5251-8-22077
juergen.binder@fujitsu-siemens.com

For Fujitsu pricing please contact:

John Fowlkes
+1-(1) 408-992-3239
John_Fowlkes@ftsi.fujitsu.com

VERITAS

Sales Quotation222052

Creation Date 10/14/2003

VERITAS Software

350 Ellis Street
P.O. Box 7011
Mountain View CA 94043-2237
800-327-2232

	To:		Account Representative:	Inside Sales Rep:
Name	RUBYE CERVELLI	**Name**	JULIAN SUAREZ	MICHAEL MAWSON
Company	FUJITSU TECHNOLOGY	**Company**	VERITAS Software	VERITAS Software
Phone	(408) 746-7926	Address	400 International ParkwayHeathrow, FL 32746	800 International ParkwayHeathrow, FL 32746
Fax				
Email	RUBYE_CERVELLI@FTSI	**Phone**	+1 (303) 383-4021	(407) 357-5193
		Fax		(407) 357-7770
		Email	julian.suarez@veritas.com	mike.mawson@veritas.com

I am pleased to accommodate your request for quotation. If you have any questions related to the products listed or require more information, please contact me at my direct telephone listed above. Thank you for your interest in VERITAS Software and I look forward to serving you in the future.

Ordering Information

(All prices are quoted in US Dollars)

Line	Qty	Part Number	Description	List Price	Ext Price
1	3	W08974F-M00212	Volume Manager,Solaris,v3.5 Tier 4C Extended Support, 1 Yr 24x7 ***Comments**Support for line 4.*	13,845.00	35,304.75
2	1	N09665F	Storage Solutions,Solaris,v3.5 Maintenance Pack 1,English,Media Kit	100.00	65.00
3	1	N08836F	Volume Manager,Solaris,Administrator's Guide,v3.5,English,Manual	50.00	32.50
4	1	A08974F-M00000	Volume Manager,Solaris,v3.5,License Tier 4C	60,195.00	39,126.75

Subtotal Information				
	License	39,126.75	**Total**	74,529.00
	Support	35,304.75		
	Finished Good	97.50		

Comments

Please note that all prices quoted are inUS Dollars and are good for 90 days form the date shown above.

Any extension of this sales quotation will only be valid when given in writing by VERITAS Software ("VERITAS").

This quotation is made subject to VERITAS standard Software License Terms and Conditions including FOB VERITAS facilities, freight prepay and charge, and payment terms of Net 30 days. Changes to such Terms and Conditions must be accepted in writing by VERITAS. Customers shall pay VERITAS a license for each software product as quoted. Customer agrees that these fees do not include any maintenance charge, any rights to additional software, enhancements, future updates and upgrades or other software products or deliverables which may be available from VERITAS. Fees for such items will be identified in the then current VERITAS price list. Any orders placed pursuant to this quotation, and any and all use of VERITAS products or services, are subject to the terms and conditions of the applicable VERITAS agreement(s) governing the particular type of order transaction involved.

Consulting

Unless otherwise specified, all VERITAS Consulting Services prices are exclusive of travel, lodging and meal expenses which will be billed at actual cost.

Tax Information

All orders are subject to sales tax. If you are requesting exemption, please provide a copy of your Sales & Use Tax Exemption Certificate.

Address Information

Bill to (For purchase orders only)	**Ship to**	**End User**
FUJITSU TECHNOLOGY SOLUTIONS, 250 EAST CARIBBEAN DR. SUNNYVALE, CA 94089 United States Attn: RUBYE CERVELLI	FUJITSU TECHNOLOGY SOLUTIONS, INC. 250 EAST CARIBBEAN DR. SUNNYVALE, CA 94089 United States Attn: RUBYE CERVELLI	FUJITSU TECHNOLOGY SOLUTIONS, INC. 250 EAST CARIBBEAN DR. SUNNYVALE, CA 94089 United States RUBYE CERVELLI

Fujitsu Siemens Computers GmbH
Heinz-Nixdorf Ring 1, 33106 Paderborn

Datum	2003-10-08
Name	Jürgen Binder
Abteilung	FSC EP ST OL
Telefon	+49 5251 8 22077
Telefax	+49 5251 8 33322077
E-mail	juergen.binder@fujitsu-siemens.com
Unser Zeichen	tpc-h_s80

Quotation

Storage

	Partnumber	Description	Quantity	Unit Price	Extended Price	Maintenance 3 years
1	D:S80-Base	FC-S80 Basis Shelf	64	3.650,00	233.600	225.000
2	D:S80FC-RDM	S80 RAID Controller 2 Gbit/s	64	9.675,00	619.200	80.000
3	D:S80-HD7310	S80 Disk, 73GB 10.000rpm	640	1.700,00	1.088.000	
4	D:GPRAC-BG52	PRIMECENTER Rack 38 HE	6	2.350,00	14.100	
				Total list	1.954.900	305.000
				Discount	879.705	
				Total	1.075.195	305.000

Comments

- Please note that all prices quoted are in US Dollars and are good for 90 days form the date shown above.

Mit freundlichen Grüßen/Best regards
Fujitsu Siemens Computers GmbH

Jürgen Binder
Sr. Product Manager Online Systems

Fujitsu Siemens Computers GmbH
Geschäftsführung: Peter Eßer (Vorsitzender), Kai Flore, Heribert Göggerle, Peter Jilek, Ulrich Kemp, Joachim Ries
Aufsichtsrat: Dr. Adrian v. Hammerstein (Vorsitzender), Stellvertreter: Otto Müller
Bankverbindung: Dresdner Bank AG (Filiale Bad Homburg), BLZ 500 800 00, Konto: 704498700; USt-IdNr.: DE113580069
Sitz der Gesellschaft: München; Registergericht: AG München, HRB 113308