TPC Benchmark™ D
Full Disclosure Report

Amdahl EnVista Frontline Server
Using
Informix-OnLine XPS 8.20TC1
and
Microsoft Windows NT 4.0

First Edition
Submitted for review
12 August 1997

First Edition -- 21 August 1997

Informix Software, Inc. and Amdahl Corporation, the Sponsors of this benchmark test, believe that the information in this document is accurate as of the publication date. The information in this document is subject to change without notice. The Sponsors assume no responsibility for any errors which may appear in this document.

The pricing information in this document is believed to accurately reflect the current prices as of the publication date. However, the Sponsors provide no warranty of the pricing information in this document.

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

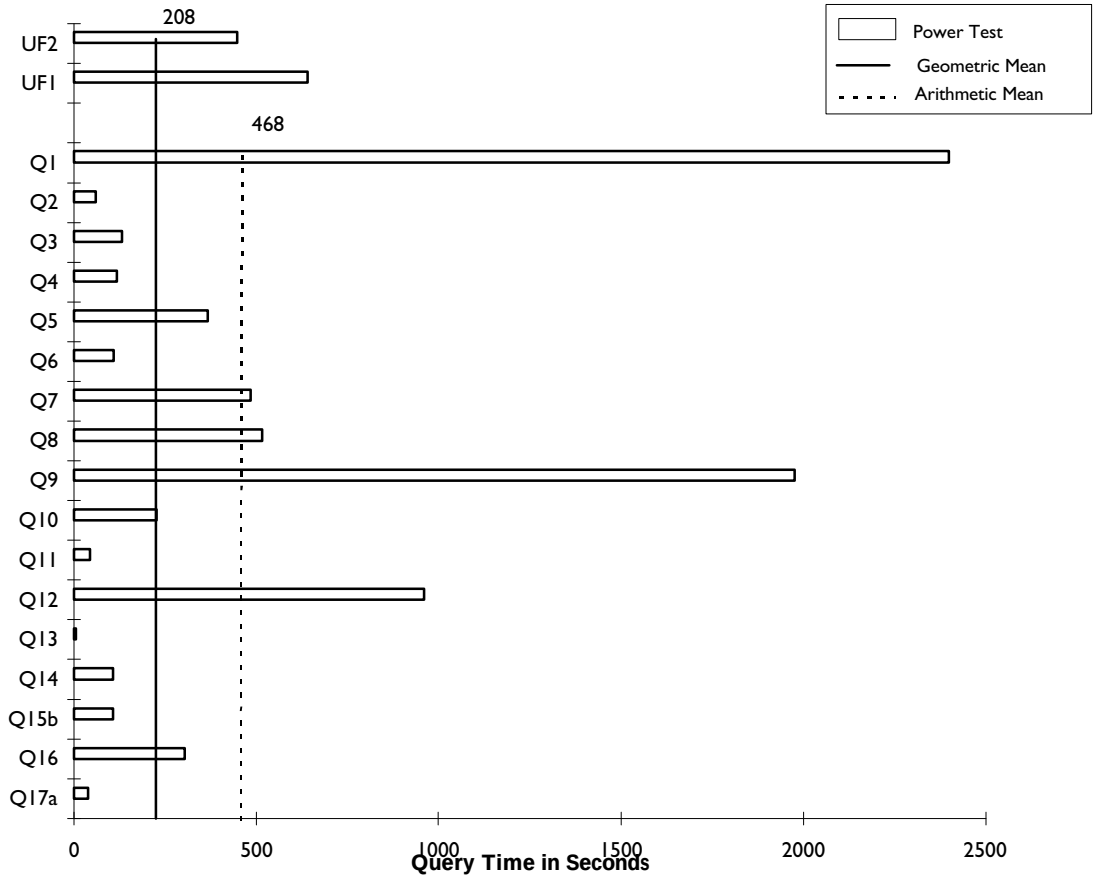
All performance data contained in this report was obtained in a rigorously controlled environment. Results obtained in other operating environments may vary significantly. No warranty of system performance or price/performance is expressed or implied in this report.

Printed in U.S.A, August 21, 1997.

INFORMIX / amdahl	**Amdahl EnVista Frontline Server Using Informix Online XPS v. 8.20TC1**	TPC-D Rev: 1.2.3 Report Date: 29-Jul-97

Total System Cost	TPC-D Power	TPC-D Throughput	Price/Performance
$607,510	**518.8** **QppD@30GB**	**200.1** **QthD@30GB**	**$1,886** **QphD@30GB**

Database Size	Database Manager	Operating System	Other Software	Availability Date
30GB	**Informix Online XPS v. 8.20TC1**	**Microsoft Windows NT 4.0**	**None**	**22-Jan-98**



Database Load Time = 9:44:27	Total Data Storage / Database Size 7.85	RAID: Y

System Configuration

4 200MHz Pentium Pro Processors w/ 512 KB cache
2GB Main Memory
62 4GB Disk Drives

Amdahl EnVista Frontline Server Using Informix Online XPS v. 8.20TC1

TPC-D Rev: 1.2.3

Report Date: 29-Jul-97

Description	Part Number	Source	Unit Price	Qty	Extended Price	5 Yr. Maint. Cost
Server Hardware						
EnVista Server Model FSR	NVISTA-FSR-4006	Amdahl	8,250	1	8,250	3,120
CPU board with 2-200MHz/512KB CPUs	NVISTA-FSR-2672	Amdahl	5,660	2	11,320	1,920
512 MB memory kit	NVISTA-FSR-2679	Amdahl	7,890	4	31,560	16,080
UltraWide SCSI Adapters	NVISTA-SCSI-2691	Amdahl	700	4	2,800	1,200
Differential SCSI Cables	NVISTA-FSR-2761	Amdahl	135	4	540	-
4.3 GB internal disks	NVISTA-FSR-2658	Amdahl	990	2	1,980	960
72" Data Center Cabinet	NVISTA-EVCAB-2747	Amdahl	3,150	1	3,150	3,000
4 mm DAT Tape Drive	NVISTA-TAPE-2704	Amdahl	1,430	1	1,430	480
Keyboard/Mouse	NVISTA-KBD-2722	Amdahl	115	1	115	-
15" Monitor	NVISTA-MON-2720	Amdahl	525	1	525	600
Keyboard/Monitor/ Mouse Extension Cables	NVISTA-EVCAB-2746	Amdahl	110	1	110	-
UPS	NVISTA-EVCAB-2738	Amdahl	3,500	1	3,500	3,000
10/100baseT Ethernet Adapter	NVISTA-ENET-2683	Amdahl	100	1	100	300
		Subtotal			**65,380**	**30,660**
Server Software						
NT 4.0 Serverw/10 CAL	NVISTA-NTE-0109	Amdahl	809	1	809	4,045
INFORMIX XPS 8.20TC1(40 users)		Informix	3,000	40	120,000	133,600
	Informix Discount		29%		(34,800)	(38,744)
Subtotal					**86,009**	**98,901**
Storage						
LVS 4500 Logical Storage Module	4500-001-8120	Amdahl	-	2	-	38,400
Intelligent Memory (96 MB kit)	4500-001-7381	Amdahl	3,995	2	7,990	-
4.3GB disk drives (20-count)	4500-001-7405	Amdahl	$77,920	2	155,840	-
Storage Expansion Module	4500-001-7410	Amdahl	$3,945	2	7,890	9,600
4.3GB disk drives (10-count)	4500-001-7395	Amdahl	$53,420	2	106,840	-
Subtotal					**278,560**	**48,000**
			Total		**429,949**	**177,561**
			5 Year Cost of Ownership		$ 607,510	
			QphD@30GB		322	
			$/QphD@30GB		**$ 1,886**	

Audited By: Francois Raab, Information Paradigm

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

Measurement Results

Scale Factor	=	30
Total Data Storage / Database Size	=	7.85
Database Load Time	=	9:44:27
Query Streams for Throughput Test	=	0
TPC-D Power Metric(QppD@30GB)	=	518.8
TPC-D Throughput Metric(QthD@30GB)	=	200.1
Composite QphD@30GB	=	322.2
Total System Price Over 5 Years	=	607,510
TPC-D Price/Performance Metric	=	$ 1,886

Measurement Intervals

Measurement Interval in Throughput Test (Ts)	=	9176

Duration of stream execution

Stream ID	Seed	Start-Date	Start-Time	End-Date	End-Time	Total Time
Stream00	1696584665	07/27/97	3:46:55	07/27/97	6:19:51	2:32:56
Updates	UF1	07/27/97	3:46:55	07/27/97	3:57:41	0:10:46
	UF2	07/27/97	6:10:10	07/27/97	6:19:51	0:09:41

TPC-D Timing Intervals (in seconds):

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
2397.2	59.3	132.2	117.2	368.1	108.5	484.6	515.0	1980.8	229.4

Q11	Q12	Q13	Q14	Q15b	Q16	Q17a	UF1	UF2
44.5	953.4	5.3	106.8	105.7	302.7	39.0	645.9	580.4

Information Paradigm

TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL

Certified Auditor

Test Sponsors: John M. Stephens, Jr.
Informix Software, Inc.
4100 Bohannon Drive
Menlo Park, CA 94025

John Howell
Amdahl Corporation
1250 E. Arques Ave.
Sunnyvale, CA 94088

July 29, 1997

I verified the TPC Benchmark D performance of the following configuration:

Platform: Amdahl EnVista Frontline Server
DataBase Manager: INFORMIX-OnLine XPS Version 8.20TCI
Operating System: Microsoft Windows NT 4.0

The results were:

CPU's Speed	Memory	Disks	QppD@30GB	QthD@30GB
Amdahl EnVista Frontline Server				
4 Pentium Pro (200 MHz)	2 GB	62 x 4 GB	518.8	200.1

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

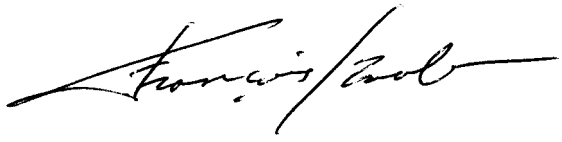
- The TIME table was not used
- The input variables were generated by QGEN
- The database was populated using DBGEN
- The database was maintained by the "Reset" method
- The throughput metric was computed using the results from the power test
- The ratio between the longest and the shortest query was such that no adjustment was necessary
- A compliant implementation specific layer was used

- The query text was produced using compliant variants and minor modifications
- The database records were defined with the proper layout and size
- The database was properly scaled to 30GB and populated accordingly
- The database load time was correctly measured and reported
- The ACID Properties were verified and met
- The reported execution times were correctly measured and reported
- Measurement repeatability was verified
- At least 8 hours of database log was configured
- The system pricing was verified for major components and maintenance
- The major pages from the FDR were verified for accuracy

Additional Audit Notes:

None.

Respectfully Yours,



Fran ois Raab
President

Amdahl EnVista Frontline Server (4-CPU)

Document Structure

The TPC Benchmark D Standard Specification requires test sponsors to publish, submit to the TPC, and make available to the public, a full disclosure report for any result considered to be compliant with the specification. The required contents for the full disclosure report are specified in Clause 8.

This report is submitted to satisfy the specification s requirement for full disclosure. It documents the compliance of the benchmark implementation and execution reported for the Amdahl EnVista Frontline Server using INFORMIX-OnLine XPS version 8.20TC1.

In the specification, the main headings in Clause 8 are keyed to the other clauses. The headings in this report use the same sequence, so that they correspond to the titles or subjects referred to in Clause 8.

Each section in this report begins with the text of the corresponding item from Clause 8 of the specification, printed using italic type. The following plain text explains how this benchmark complies with that specific portion of the specification. In sections where Clause 8 requires extensive listings the appropriate appendix in this report is referenced.

TPC Benchmark™ D Overview

The TPC Benchmark D (TPC-D) is a decision support benchmark. It is a suite of business oriented queries and concurrent updates. The queries and the data populating the database have been chosen to have broad industry-wide relevance while maintaining a sufficient degree of ease of implementation. The 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-D evaluates the performance of various decision support systems by the execution of sets of queries against a standard database under controlled conditions. The TPC-D queries:

- Give answers to real-world business questions;
- Are far more complex that most OLTP transactions;
- Include a rich breadth of operators and selectivity constraints;
- Generate intensive activity on the part of the database server components 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.

Table of Contents

List of Tables and Diagrams

1 General Information

1.1 Benchmark Sponsor

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

Informix Software, Inc. and Amdahl Corporation are the sponsors of this TPC-D benchmark.

1.2 Parameter Settings

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

Database tuning options;

Optimizer/query execution options;

Query processing tool/language configuration parameters;

Recovery/commit options;

Consistency/locking options;

Operating system and configuration parameters;

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

Compiler optimization options

This requirement can be satisfied by providing a full list of all parameters and options.

Appendix A,

DBMS and System Parameters , contains the XPS and Microsoft Windows NT 4.0 parameters used in this benchmark.

1.3 Configuration Diagrams

Provide diagrams of both the measured and priced configurations, accompanied by a description of the differences. This includes, but is not limited to:

Number and type of processors;

Size of allocated memory, and any specific mapping/partitioning of memory unique to the test;

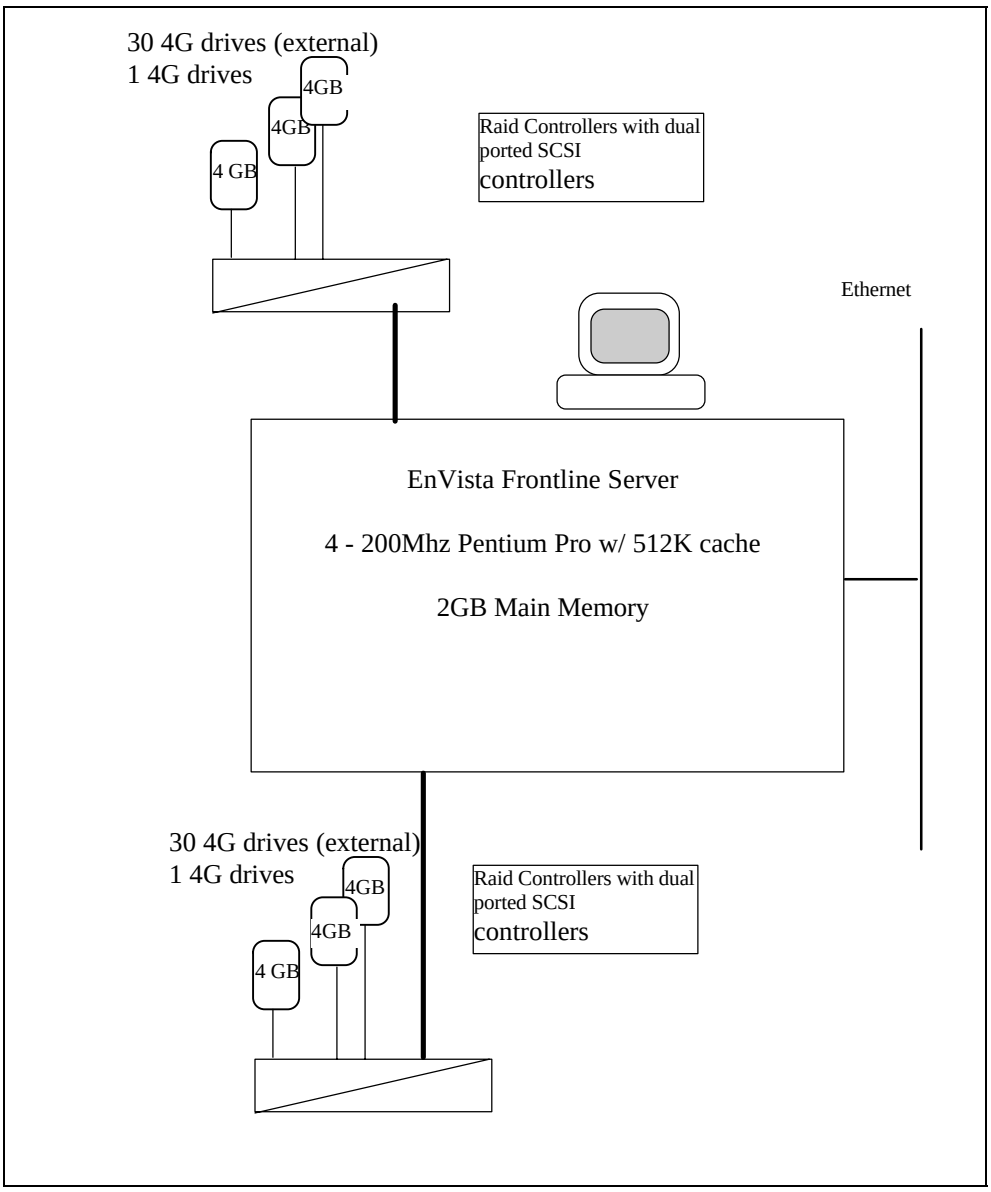
Number of type of disk unites (and controllers, if applicable);

Number of channels or bus connections to disk units, including their protocol type;

Number of LAN (e.g., Ethernet) connections, including routers, workstations, terminals, etc., that were physically used in the test or are incorporated into the pricing structure;

Type and run-time execution location of software components (e.g., DBMS, query processing tools/languages, middle-ware components, software drivers, etc.)

The priced and tested configurations were identical, consisting of a single Amdahl Corporation Amdahl EnVista Frontline Server, configured with 4 Pentium-Pro processors running at 200Mhz and including 512KB cache. The system contained 2GB of main memory, 2 RAID modules, each with 2 SCSI controllers, and a total of 60 9GB disk drives and 2 4G disk drives. All 60 9GB disks were mirrored at the controller level to assure data integrity.



2 Clause 1 Logical Database Design

2.1 Database Definition Statements

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

Appendix B, Database Creation Statements , contains the programs and scripts that create and analyze the tables and indexes for the qualification and test databases used in this benchmark.

2.2 Physical Organization

The physical organization of tables and indexes, 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 in this benchmark. No changes were made to the column ordering presented in Clause 1.4 of the specification. Further details can be found in Appendix B, Database Creation Statements .

2.3 Horizontal Partitioning

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

Horizontal partitioning was used for all tables and indexes except nation and region. Further details can be found in Appendix B, Database Creation Statements .

2.4 Replication

While there are some restrictions placed upon the physical replication of objects in the test and qualification database (see Clause 1.5.6), any such replication must be disclosed.

No replication was used in this benchmark.

3 Clause 2 Queries and Update Functions

3.1 Query Language

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

INFORMIX-SQL was used to implement the queries and update functions used in this benchmark.

3.2 Random Number Generation

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

This benchmark used version 1.2.0 of DBGEN and QGEN without modification for random number generation.

3.3 Substitution Parameter Generation

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

This benchmark used version 1.2.0 of QGEN. The only modification was to modify the text used to control the number of rows returned to conform to Informix s SELECT FIRST <n> syntax. The change required in tpcd.h is summarized below.

OLD: #define SET_ROWCOUNT {return %d rows}\n

NEW: #define SET_ROWCOUNT FIRST %d

3.4 Query Text and Output Data used for Query Validation

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 query modifications have been applied to any functional query definition or approved variant in order to obtain executable query text, these modifications must be disclosed and justified. The justification for a particular minor query modification can be applies collectively to all queries for which it has been used.)

Appendix C, Query Validation EQT and Output , contains the executable query text used during query validation and the resulting output. The following minor query modifications were used to obtain the executable query text that was used for query validation and the performance runs:

- In Q2, Q3 and Q10 the Informix SELECT FIRST <N> was used to limit the size of the answer set as allowed by Clause 2.1.2.7.

- All date expressions were rewritten using equivalent INFORMIX syntax as allowed by Clause 2.2.3.3c.

- In Q7, Q8, Q9 and Q13, the nested table expression included in the functional query definition solely for the purpose of grouping on an expression have been removed, table names promoted from the nested table expression to the from clause, and the GROUP BY and ORDER BY clauses modified to use an ordinal in place of the nested table expression referenced in the functional query definitions. This is allowed by Clause 2.2.3.3d.

- In Q8, ROUND(, 2) has been added to the outermost select to control intermediate arithmetic precision as allowed by 2.2.3.3f.

- In Q15b and Q17a, the space allocation directive in <dbslice> has been added to the table creation syntax provided in the approved query variant as allowed by 2.2.3.3j.

3.5 Substitution Parameters and QGEN Seeds

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, Substitution Parameters and Seeds , includes the QGEN seed value and resulting substitution parameters used in the performance tests.

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 update functions were run at isolation level 2 (as defined in Clause 3.4), what Informix refers to as Repeatable Read.

3.7 Source Code of Update Functions

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

Appendix E, Implementation Specific Layer and Drivers , contains the full source code for the update functions and their associated reset routines.

3.8 Database Maintenance Option

The details of the database maintenance option selected (i.e., reset or evolve) must be disclosed (including the source code of any non-commercial program used).

This benchmark used the reset option. Source code for the associated scripts can be found in Appendix E, Implementation Specific Layer and Drivers .

4 Clause 3 Database System Properties

4.1 ACID Properties

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

Complete source code for the ACID test is included in Appendix F, ACID Test Source Code .

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 ORDER, LINEITEM, and HISTORY tables.

1. The total price from the ORDER table and the extended price from the LINEITEM table were retrieved for a randomly selected order key.

2. The ACID Transaction was performed using the order key from Step 1.

3. The ACID Transaction committed.

4. The total price from the ORDER table and the extended price from the LINEITEM table were retrieved for the same order key. It was verified that the appropriate rows had been changed.

4.2.2 Aborted Transaction

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

1. The total price from the ORDER table and the extended price from the LINEITEM table were retrieved for a randomly selected order key.

2. The ACID Transaction was performed using the order key from Step 1. The transaction was stopped prior to the commit.

3. The COMMIT was replaced with a ROLLBACK.

4. The total price from the ORDER table and the extended price from the LINEITEM table were retrieved for the same order key. It was verified that the appropriate rows had 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 the ORDER 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 ORDER and LINEITEM tables.

1. The consistency of the ORDER and LINEITEM tables was verified based on a sample of O_ORDERKEY s.

2. 100 ACID Transactions were submitted from each of 2 execution streams.

3. The consistency of the ORDER and LINEITEM tables was re-verified.

4.4 Isolation

Operation of concurrent transactions must yield results which are indistinguishable from the result which would have been 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 with 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 did not see the uncommitted changes made by the ACID Transaction.

3. The ACID Transaction resumed executed a COMMIT.

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 resumed executed a ROLLBACK WORK.

4. The ACID Query completed. It returned the data as seen prior to the start of the ACID Transaction.

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, DELTA1. 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, DELTA2.

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, DELTA1. 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, DELTA2.

3. T2 waited.

4. T1 was allowed to COMMIT 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 ACID Transaction, T2, was started and, using randomly selected values of PS_PARTKEY and PS_SUPPKEY, selected all rows in the

PARTSUPP table for which match the selected values of PS_PARTKEY and PS_SUPPKEY.

3. T2 completed.

4. T1 was allowed to COMMIT.

5. It was verified that appropriate rows in the ORDER, 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 of the tables of the database does not indefinitely delay update transactions affecting those tables from making progress.

1. A database session, S1, began an execution of Q1 against the qualification database using a randomly selected DELTA.

2. While S1 was executing, a second database session, S2, began an ACID Transaction, T1, using randomly selected values for O_KEY, L_KEY and DELTA.

3. Immediately following its first execution for Q1, S1 began another execution of Q1 using a different, randomly selected DELTA.

4. T1 waited for the first execution of Q1 by S1 to complete, and then completed its ACID Transaction. It was verified that the appropriate rows in the ORDER, LINEITEM and HISTORY tables had been changed.

5. The second execution of Q1 by S1 waited for T1 to complete and then it proceeded to completion.

4.5 Durability

The tested system must guarantee durability: the ability to preserve the effects of committed transactions and ensure database consistency after recovery from any one of the failures listed in Clause 3.5.2.

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-D database tables or recovery log files.

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

4.5.2 System Crash and Memory Failure

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

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

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

5 Clause 4 Scaling and Database Population

5.1 Table Cardinalities

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

Table	Rows
Order	45,000,000
Lineitem	179,998,372
Customer	4,500,000
Part	6,000,000
Supplier	300,000
Partsupp	24,000,000
Nation	25
Region	5

Figure 1: Table Cardinalities

5.2 Distribution of Tables and Logs Across Media

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

This benchmark used NT filesystems to hold all database files and logs, as well as all DBMS and OS executables. The allocation of disk to the various tasks is detailed in the table below.

	DISK		
Drive	Count	Size	Use
C	0.5	4GB	NT System
D	0.5	4GB	Informix Tools
E			CD-ROM
F	1	4GB	
G -T	4	4GB	1/14th of Data/Index/Temp
W	2	4GB	ROOTDBS, Physical Log, Logical Log

Figure 2: Disk Usage Summary

5.3 Database Partition/Replication Mapping

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

The database used in this benchmark employed no replication. For a detailed disclosure of the creation and use of database partitions, refer to Appendix B, Database Creation Statements .

5.4 RAID Usage

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.

RAID 1 (i.e., mirroring) was used for all database tables, logs and indexes, including the root dbspace.

5.5 Modifications to DBGEN

Any modification 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 1.2.0) was used without modification.

5.6 Database Contents Validation

The contents of the first ten rows of each table in the test database must be disclosed.

Appendix G, Database Contents contains the first 10 rows of each table in the test database.

5.7 Database Load Time

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

The database load time was 9:44:27.

5.8 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 database size as defined in Clause 4.1.3.

The data storage ratio reported in the Executive Summary in the Preface to this report was based on the following information:

Disk Type	# of Disks	Disk Capacity	Disk Space
4 GB (ext)	60	3.8 GB	228 GB
4 GB (int)	2	3.8 GB	7.6 GB
		Total:	235.6 GB
		SF:	30 GB
Data Storage Ratio			**7.85**

Figure 3: Data Storage Ratio Details

5.9 Database Load Description

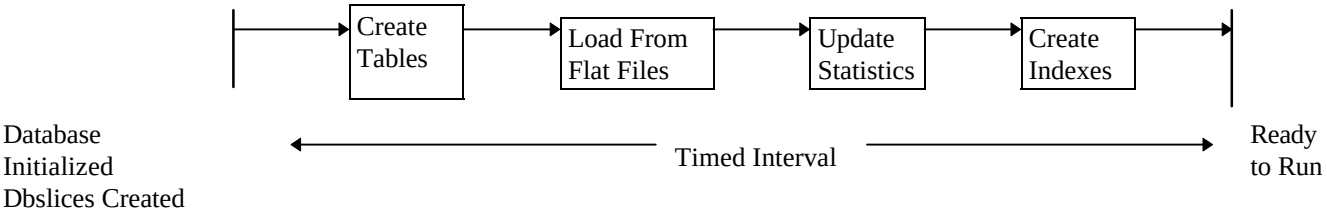
The details of the database load mechanism must be described and illustrated with a block diagram.

The database was loaded using XPS/Pload, which used flat files generated by DBGEN as input. The same mechanism was used to load both the qualification database and the test database. The number of flat files used to load each table within each data set is detailed below, along with a depiction of the complete load process. All scripts and command files used during the load are contained in Appendix B, Database Creation Statements .

Table	Database	
	Qualification	Test
Order	1	80
Lineitem	1	80
Customer	1	80
Part	1	80
Supplier	1	80
Partsupp	1	80
Nation	1	1
Region	1	1

Figure 4: Flat File Usage Summary

Figure 5: Load Test Process Summary



6 Clause 5 Performance Metrics and Execution Rules

6.1 Power Test Overview

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

The following procedure was used to implement the power test:

1. Database Restart
2. UF1 Update Function Executed
3. Query Stream 00 Executed
4. UF2 Update Function Executed

6.2 Power Test Timing Intervals

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

The timing intervals for the reported power test were:

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
2397.2	59.3	132.2	117.2	368.1	108.5	484.6	515.0	1980.8	229.4

Q11	Q12	Q13	Q14	Q15b	Q16	Q17a	UF1	UF2
44.5	953.4	5.3	106.8	105.7	302.7	39.0	645.9	580.4

Figure 6: Power Test Timing Intervals

6.3 Throughput Test Configuration

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

The throughput test was not executed during this benchmark. The timing intervals from the power test were used to calculate the throughput metric as allowed by Clause 5.3.1.4.

6.4 Query Stream Timings

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

The start and end time for each query stream are contained in the Numerical Quantities Summary of the Executive Summary in the preface of this report.

6.5 Elapsed Time of the Measurement Interval

The total elapsed time of the measurement interval must be disclosed.

The total elapsed time of the measurement interval is contained in the Numerical Quantities Summary of the Executive Summary in the preface of this report.

6.6 Timing Intervals for the Throughput Test

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

Since this benchmark did not execute the throughput test, the timing intervals for the power test were used to calculate the throughput metric. The timing intervals for each query and update function in the power test are contained in the Numerical Quantities Summary of the Executive Summary in the preface of this report.

6.7 Performance Metrics

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

The performance metrics, the price performance metric and all the underlying data on which they are based are contained in the Numerical Quantities Summary of the Executive Summary in the preface of this report.

6.8 Metric Reproducibility

A description of the method used to determine the reproducibility of the measurements must be reported. This must include the performance metrics (QppD, QthD and QphD) from the reproducibility runs.

Performance results from consecutive runs of the performance test revealed the following variability in the performance metrics:

	Run1	Run2	Run3	Reported
QppD @ 30 GB	526.1	518.8	519.9	518.8
QthD @ 30 GB	203.0	200.1	199.6	200.1
QphD @ 30 GB	326.8	322.2	322.1	322.2
$/QphD @ 30 GB				$1,886
Variation	1.44%			
	1.47%			

Figure 7: Metric Variability

7 Clause 6: SUT and Driver Implementation

7.1 Driver Overview

A detailed textual description of how the driver performs its functions, how its various components interact and any product functionality or environmental setting on which it relies must be disclosed.

The driver configuration used for this implementation of TPC-D is quite simple. A central script(driver) relies on QGEN to translate query templates into Executable Query Text(EQT). These are submitted to the DBMS through Informix s standard interactive query tool, DBACCESS. The update functions (and their associated reset scripts) use standard Informix SQL to load from the flat files generated by DBGEN. The rows are either loaded directly into the base tables(for UF1 and uf2_reset) or are loaded into a temporary table which is then used to direct the deletion of rows from the base tables (for UF2 and uf1_reset).

The complete source code for the driver script and all related non-commercial tools is provided in Appendix E, Implementation Specific Layer and Drivers .

7.2 Implementation Specific Layer Overview

If an implementation specific layer is used, then a detailed textual description of how the driver performs its functions, how its various components interact and any product functionality or environmental setting on which it relies must be disclosed.

Since standard Informix tools provide all the functionality needed to execute the benchmark, no implementation specific layer was required.

7.3 Profile-directed Optimization

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

No profile-directed optimization was used in this benchmark.

8 Clause 7: Pricing

8.1 Pricing Summary

A detail list of hardware and software used in the priced system must be reported.

Please refer to the Pricing Spreadsheet found in the executive summary at the beginning of this report.

8.2 Total System Cost

The total 5-year price of the entire configuration must be reported, including hardware, software and maintenance charges.

Please refer to the Pricing Spreadsheet found in the executive summary at the beginning of this report.

8.3 System Availability

The committed delivery date for general availability (availability date) of products used in the priced calculations must be reported.

Please refer to the executive summary at the beginning of this report.

9 Clause 9: Audit

9.1 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.

The attestation letter for this benchmark is included at the beginning of this report.

Appendix A: DBMS and System Parameters

A-1: DBMS Parameters

Note: The following is a complete listing of the ONCONFIG used on the SUT. Since all but a few settings are system dependent, and therefore differ from the defaults, highlighting of changes has been omitted for easier reading.

```
#**************************************************************************
#
#                    INFORMIX SOFTWARE, INC.
#
#  Title:       onconfig.std-> xps (04.01.96, gws).
#  Description: INFORMIX-OnLine Configuration Parameters
#
#**************************************************************************

# Root Dbspace Configuration

ROOTOFFSET      0               # Offset of root dbspace into
device (Kbytes)
ROOTSIZE                2024000                 # Size of root
dbspace (Kbytes)

# Disk Mirroring Configuration Parameters

MIRROR          0               # Mirroring flag (Yes = 1, No = 0)
MIRRORPATH                      # Path for device containing
mirrored root
MIRROROFFSET    0               # Offset into mirrored device
(Kbytes)

# Physical Log Configuration

PHYSFILE        1900000         # Physical log file size
(Kbytes)
PHSYSLICE       rootdbs

# Logical Log Configuration

LOGFILES        5               # Number of logical log files
LOGSIZE         10000           # Logical log size (Kbytes)

# Diagnostics

MSGPATH \\inform2\e$\informix\online.log
CONSOLE \\inform2\e$\informix\console.log
ALARMPROGRAM                            # Alarm program
path

# System Archive Tape Device

TAPEDEV         nul             # Tape device path
TAPEBLK         16              # Tape block size (Kbytes)
TAPESIZE        10240           # Maximum amount of data to
put on tape (Kbytes)

# Log Archive Tape Device

LTAPEDEV        nul             # Log tape device path
LTAPEBLK        16              # Log tape block size (Kbytes)
LTAPESIZE       10240           # Max amount of data to put on
log tape (Kbytes)

# Optical

STAGEBLOB                       # INFORMIX-
OnLine/Optical staging area

# System Configuration

SERVERNUM       10              # Unique id corresponding to a
OnLine instance
DBSERVERNAME    ol_inform2      # Name of default
database server
DBSERVERALIASES         # List of alternate dbservernames
NETTYPE         onsoctcp,1,,NET # Override sqlhosts nettype
parameters
DEADLOCK_TIMEOUT        60      # Max time to wait of
lock in distributed env.
RESIDENT        1               # Forced residency flag (Yes = 1,
No = 0)

MULTIPROCESSOR  1               # 0 for single-processor, 1
for multi-processor
NUMCPUVPS       4                       # Number of user (cpu)
vps
SINGLE_CPU_VP   0               # If non-zero, limit number of
cpu vps to one

NOAGE           0               # Process aging
AFF_SPROC       0               # Affinity start processor
AFF_NPROCS      0               # Affinity number of processors

# Shared Memory Parameters

USERTHREADS     100
TRANSACTIONS    100
LOCKS           900000          # Maximum number of locks
BUFFERS         120000          # Maximum number of shared
buffers
TBLSPACES       1600            # Maximum number of open
tblspaces
CHUNKS          1600            # Maximum number of chunks
NUMAIOVPS       8               # Number of IO vps
DBSPACES        1600            # Maximum number of
dbspaces
PHYSBUFF        64              # Physical log buffer size (Kbytes)
LOGBUFF         64              # Logical log buffer size (Kbytes)
LOGSMAX         40              # Maximum number of logical
log files
CLEANERS        32              # Number of buffer cleaner
processes
#SHMBASE        0x20000000L     # Shared memory base
address
SHMBASE         0xC000000L      # Shared memory base
address
SHMVIRTSIZE     900000          # initial virtual shared
memory segment size
#SHMVIRTSIZE    8192            # initial virtual shared
memory segment size
SHMADD          81920           # Size of new shared memory
segments (Kbytes)
# SHMADD        420000          # Size of new shared memory
segments (Kbytes)
SHMTOTAL        1700000   # Total shared memory (Kbytes).
0=>unlimited
CKPTINTVL       3600            # Check point interval (in sec)
LRUS            32              # Number of LRU queues
LRU_MAX_DIRTY   70              # LRU percent dirty begin
cleaning limit
```

```
LRU_MIN_DIRTY   60          # LRU percent dirty end
cleaning limit
LTXHWM          50          # Long transaction high water
mark percentage
LTXEHWM         60          # Long transaction high water
mark (exclusive)
TXTIMEOUT       300         # Transaction timeout (in sec)
STACKSIZE       32          # Stack size (Kbytes)

# System Page Size
# BUFFSIZE - OnLine no longer supports this configuration
parameter.
#         To determine the page size used by OnLine on your
platform
#         see the last line of output from the command, 'onstat -
b'.


# Recovery Variables
# OFF_RECVRY_THREADS:
# Number of parallel worker threads during fast recovery or an
offline restore.
# ON_RECVRY_THREADS:
# Number of parallel worker threads during an online restore.

OFF_RECVRY_THREADS      10      # Default number of
offline worker threads
ON_RECVRY_THREADS       1       # Default number of
online worker threads

# Data Replication Variables
# DRAUTO: 0 manual, 1 retain type, 2 reverse type
DRAUTO          0           # DR automatic switchover
DRINTERVAL      30          # DR max time between DR
buffer flushes (in sec)
DRTIMEOUT       30          # DR network timeout (in sec)
DRLOSTFOUND     e:\tmp      # DR lost+found file path


# Read Ahead Variables
RA_PAGES        128         # Number of pages to attempt to
read ahead
RA_THRESHOLD    64          # Number of pages left
before next group

# DBSPACETEMP:
# OnLine equivalent of DBTEMP for SE. This is the list of
dbspaces
# that the OnLine SQL Engine will use to create temp tables
etc.
# If specified it must be a colon separated list of dbspaces that
exist
# when the OnLine system is brought online.  If not specified,
or if
# all dbspaces specified are invalid, various ad hoc queries will
create
# temporary files in /tmp instead.

#DBSPACETEMP    ALL         # Default temp dbspaces
DBSPACETEMP
temp_slice1,temp_slice2,temp_slice3,temp_slice4,temp_slice5,
temp_slice6,temp_slice7,temp_slice11,temp_slice12,temp_slic
e13,temp_slice14,temp_slice15,temp_slice16,temp_slice17,roo
tdbs     # Default temp dbspaces

# DUMP*:
# The following parameters control the type of diagnostics
information which
# is preserved when an unanticipated error condition (assertion
failure) occurs
# during OnLine operations.
# For DUMPSHMEM, DUMPGCORE and DUMPCORE 1
means Yes, 0 means No.

DUMPDIR \tmp                # Preserve diagnostics in this
directory
DUMPSHMEM       1           # Dump a copy of shared
memory
DUMPGCORE       0           # Dump a core image using
'gcore'
DUMPCORE        0           # Dump a core image
(Warning:this aborts OnLine)
DUMPCNT 1                   # Number of shared memory or
gcore dumps for
                                        # a single user's
session

# ADT*
# ADT* parameters moved to adtcfg file

FILLFACTOR      95      # Fill factor for building indexes

# method for OnLine to use when determining current time
USEOSTIME       0       # 0: use internal time(fast), 1: get time
from OS(slow)

# Parallel Database Queries (pdq)
PDQPRIORITY     100     # Degree of parallelism: 0 ... 100,
                                #  OFF => 0, LOW => 1,
HIGH => 100
MAX_PDQPRIORITY 100     # Maximum allowed
pdqpriority
DS_MAX_QUERIES  1       # Maximum number of decision
support queries
DS_TOTAL_MEMORY 1000000     # Decision support
memory (Kbytes)
DS_MAX_SCANS    28      # Maximum number of decision
support scans
DS_POOLSIZE     500
DS_HASHSIZE     251
DATASKIP            # List of dbspaces to skip

# OPTCOMPIND
# 0 => Nested loop joins will be preferred (where
#      possible) over sortmerge joins and hash joins.
# 1 => If the transaction isolation mode is not
#      "repeatable read", optimizer behaves as in (2)
#      below.  Otherwise it behaves as in (0) above.
# 2 => Use costs regardless of the transaction isolation
#      mode.  Nested loop joins are not necessarily
#      preferred.  Optimizer bases its decision purely
#      on costs.
# XSOPTCOMPIND      2       # To hint the optimizer

LOG_BACKUP_MODE NONE
CONFIGSIZE LARGE
# CONFIGSIZE HUGE

# XPS_GLOBAL_CONFIGURATION_ENDS
#XPS ADDITIONS
# ROOTPATH          E:\IFMXDATA\ol_inform2\rootdbs%c
ROOTPATH            W:\ol_inform2\rootdbs%c
ROOTSLICE       rootdbs
PHYSSLICE       rootdbs

SBUFFER 880
LBUFFER 8032
HBUFFER 48992

COSERVER        1
NODE            inform2 #
SADDR           inform2:26800,5,0
```

```
LADDR           inform2:26805,20,0
HADDR           inform2:26925,10,0

SADDR           inform2:27800,5,1
LADDR           inform2:27805,20,1
HADDR           inform2:27925,10,1

SADDR           inform2:28800,5,2
LADDR           inform2:28805,20,2
HADDR           inform2:28925,10,2

SADDR           inform2:29800,5,3
LADDR           inform2:29805,20,3
HADDR           inform2:29925,10,3
END

# COSERVER      2
# NODE          INFORM2 #
# SADDR         inform2:26800,5,0
# LADDR         inform2:26805,20,0
# HADDR         inform2:26925,10,0

# SADDR         inform2:27800,5,1
# LADDR         inform2:27805,20,1
# HADDR         inform2:27925,10,1
#
# SADDR         inform2:28800,5,2
# LADDR         inform2:28805,20,2
# HADDR         inform2:28925,10,2
#
# SADDR         inform2:29800,5,3
# LADDR         inform2:29805,20,3
# HADDR         inform2:29925,10,3
# END
```

A-2: Environment Variable Settings

```
set INFORMIXDIR=\\INFORM2\E$\Informix
set INFORMIXSERVER=ol_inform2.1
set ONCONFIG=ONCONFIG
set PATH=\\INFORM2\E$\Informix\bin;%PATH%;
set INFORMIXSQLHOSTS=\\INFORM2
set DBNLS=0
set LANG=English
set COLLCHAR=1

set DSS_CONFIG=W:\tpcd\dbgen
set DSS_SEED=W:\tpcd\seeds
set DSS_LOG=W:\tpcd\log
set DSS_BAT=W:\tpcd\bat
set DSS_BIN=W:\tpcd\bin
set
PATH=%PATH%;%DSS_CONFIG%;%DSS_BAT%;%DSS
_BIN%;C:\msdev\bin;w:\tpcd\;
set SQXPLN=E:\informix\sqexpln\informix.out
set DBDATE=Y4MD-
set PDQPRIORITY=100
set DEBUG=MAXSCAN:4,HFEVAL
```

A-3: System Parameters

```
Key Name:       SOFTWARE\Informix
Class Name:     <NO CLASS>
Last Write Time:  7/8/97 - 3:47 PM

Key Name:       SOFTWARE\Informix\ESQL/C
Class Name:     <NO CLASS>
Last Write Time:  7/16/97 - 1:09 PM
Value 0
 Name:          INFORMIX-Connect
 Type:          REG_SZ
 Data:          7.20TE1


Key Name:       SOFTWARE\Informix\ESQL/C\7.20TE1
Class Name:     <NO CLASS>
Last Write Time:  7/16/97 - 1:02 PM

Key Name:       SOFTWARE\Informix\INFORMIX-Connect
Class Name:     <NO CLASS>
Last Write Time:  7/16/97 - 1:09 PM

Key Name:       SOFTWARE\Informix\INFORMIX-
Connect\7.20TE1
Class Name:     <NO CLASS>
Last Write Time:  7/16/97 - 1:09 PM

Key Name:       SOFTWARE\Informix\INFORMIX-
Connect\7.20TE1\Setup
Class Name:     <NO CLASS>
Last Write Time:  7/16/97 - 1:09 PM
Value 0
 Name:          InstallDate
 Type:          REG_SZ
 Data:          7/16/1997

Value 1
 Name:          PathName
 Type:          REG_SZ
 Data:          e:\informix

Value 2
 Name:          RegisteredOrganization
 Type:          REG_SZ
 Data:          informix

Value 3
 Name:          RegisteredOwner
 Type:          REG_SZ
 Data:          Informix

Value 4
 Name:          SoftwareType
 Type:          REG_SZ
 Data:          RDBMS


Key Name:       SOFTWARE\Informix\RCE
Class Name:     <NO CLASS>
Last Write Time:  7/8/97 - 3:50 PM
Value 0
 Name:          group
 Type:          REG_SZ
 Data:          LOTADISK\Informix-Admin


Key Name:       SOFTWARE\Informix\Setup Framework
Class Name:     <NO CLASS>
Last Write Time:  7/8/97 - 3:47 PM

Key Name:       SOFTWARE\Informix\Setup
Framework\CurrentVersion
Class Name:     <NO CLASS>
Last Write Time:  7/8/97 - 3:47 PM

Key Name:       SOFTWARE\Informix\Setup
Framework\CurrentVersion\Setups
Class Name:     <NO CLASS>
Last Write Time:  7/8/97 - 3:47 PM
```

```
Key Name:         SOFTWARE\Informix\Setup
Framework\CurrentVersion\Setups\Directories
Class Name:       <NO CLASS>
Last Write Time:  7/8/97 - 3:47 PM
Value 0
  Name:           \\INFORM2\E$\Informix
  Type:           REG_DWORD
  Data:           0x1

Value 1
  Name:           \\INFORM2\E$\Informix\aaodir
  Type:           REG_DWORD
  Data:           0x1

Value 2
  Name:           \\INFORM2\E$\Informix\bin
  Type:           REG_DWORD
  Data:           0x1

Value 3
  Name:           \\INFORM2\E$\Informix\dbssodir
  Type:           REG_DWORD
  Data:           0x1

Value 4
  Name:           \\INFORM2\E$\Informix\demo
  Type:           REG_DWORD
  Data:           0x1

Value 5
  Name:           \\INFORM2\E$\Informix\demo\dbaccess
  Type:           REG_DWORD
  Data:           0x1

Value 6
  Name:           \\INFORM2\E$\Informix\etc
  Type:           REG_DWORD
  Data:           0x1

Value 7
  Name:           \\INFORM2\E$\Informix\forms
  Type:           REG_DWORD
  Data:           0x1

Value 8
  Name:           \\INFORM2\E$\Informix\help
  Type:           REG_DWORD
  Data:           0x1

Value 9
  Name:           \\INFORM2\E$\Informix\infxtmp
  Type:           REG_DWORD
  Data:           0x1

Value 10
  Name:           \\INFORM2\E$\Informix\msg
  Type:           REG_DWORD
  Data:           0x1

Value 11
  Name:           \\INFORM2\E$\Informix\release
  Type:           REG_DWORD
  Data:           0x1

Value 12
  Name:           \\INFORM2\E$\Informix\sqexpln
  Type:           REG_DWORD
  Data:           0x1

Value 13
  Name:           \\INFORM2\E$\tmp
  Type:           REG_DWORD
  Data:           0x2


Key Name:         SOFTWARE\Informix\Setup
Framework\CurrentVersion\Setups\Files
Class Name:       <NO CLASS>
Last Write Time:  7/8/97 - 3:50 PM
Value 0
  Name:
\\INFORM2\\\INFORM2\admin$\system32\drivers\etc\hosts.eq
uiv
  Type:           REG_DWORD
  Data:           0x2

Value 1
  Name:           \\INFORM2\E$\Informix\aaodir\adtcfg.0
  Type:           REG_DWORD
  Data:           0x1

Value 2
  Name:           \\INFORM2\E$\Informix\console.log
  Type:           REG_DWORD
  Data:           0x1

Value 3
  Name:           \\INFORM2\E$\Informix\etc\.infos.ol_inform2
  Type:           REG_DWORD
  Data:           0x1

Value 4
  Name:           \\INFORM2\E$\Informix\etc\bldutil.out
  Type:           REG_DWORD
  Data:           0x1

Value 5
  Name:           \\INFORM2\E$\Informix\etc\buildsmi.out
  Type:           REG_DWORD
  Data:           0x1

Value 6
  Name:
\\INFORM2\E$\Informix\etc\oncfg_ol_inform2.0
  Type:           REG_DWORD
  Data:           0x1

Value 7
  Name:           \\INFORM2\E$\Informix\ETC\ONCONFIG
  Type:           REG_DWORD
  Data:           0x1

Value 8
  Name:           \\INFORM2\E$\Informix\help\errmsg.ann
  Type:           REG_DWORD
  Data:           0x1

Value 9
  Name:           \\INFORM2\E$\Informix\help\errmsg.ftg
  Type:           REG_DWORD
  Data:           0x1

Value 10
  Name:           \\INFORM2\E$\Informix\help\errmsg.fts
  Type:           REG_DWORD
  Data:           0x1

Value 11
  Name:           \\INFORM2\E$\Informix\help\errmsg.gid
  Type:           REG_DWORD
  Data:           0x1
```

Value 12
Name: \\INFORM2\E$\Informix\help\errmsg.ph
Type: REG_DWORD
Data: 0x1

Value 13
Name: \\INFORM2\E$\Informix\online.log
Type: REG_DWORD
Data: 0x1

Value 14
Name: \\INFORM2\E$\Informix\SETENV.CMD
Type: REG_DWORD
Data: 0x1

Value 15
Name:
\\INFORM2\U$\IFMXDATA\ol_inform2\rootdbs1
Type: REG_DWORD
Data: 0x1

Value 16
Name: C:\TEMP\setup.log
Type: REG_DWORD
Data: 0x1

Value 17
Name: C:\WINNT\System32\online_service.log
Type: REG_DWORD
Data: 0x1

Value 18
Name: E:\Informix\bin\DBACCESS.EXE
Type: REG_DWORD
Data: 0x1

Value 19
Name: E:\Informix\bin\dbschema.exe
Type: REG_DWORD
Data: 0x1

Value 20
Name: E:\Informix\bin\dgtrans.ini
Type: REG_DWORD
Data: 0x1

Value 21
Name: E:\Informix\bin\dgtrans.sys
Type: REG_DWORD
Data: 0x1

Value 22
Name: E:\Informix\bin\ifmxdg.dll
Type: REG_DWORD
Data: 0x1

Value 23
Name: E:\Informix\bin\imacrouter.exe
Type: REG_DWORD
Data: 0x1

Value 24
Name: E:\Informix\bin\makedate.exe
Type: REG_DWORD
Data: 0x1

Value 25
Name: E:\Informix\bin\onevd.exe
Type: REG_DWORD
Data: 0x1

Value 26
Name: E:\Informix\bin\oninit.exe
Type: REG_DWORD
Data: 0x1

Value 27
Name: E:\Informix\bin\ONMODE.EXE
Type: REG_DWORD
Data: 0x1

Value 28
Name: E:\Informix\bin\ONPARAMS.EXE
Type: REG_DWORD
Data: 0x1

Value 29
Name: E:\Informix\bin\ONSPACES.EXE
Type: REG_DWORD
Data: 0x1

Value 30
Name: E:\Informix\bin\ONSTAT.EXE
Type: REG_DWORD
Data: 0x1

Value 31
Name: E:\Informix\bin\onutil.exe
Type: REG_DWORD
Data: 0x1

Value 32
Name: E:\Informix\bin\rce.exe
Type: REG_DWORD
Data: 0x1

Value 33
Name: E:\Informix\bin\rcmdsvc.exe
Type: REG_DWORD
Data: 0x1

Value 34
Name: E:\Informix\bin\readme.txt
Type: REG_DWORD
Data: 0x1

Value 35
Name: E:\Informix\bin\xboot.exe
Type: REG_DWORD
Data: 0x1

Value 36
Name: E:\Informix\bin\xctl.exe
Type: REG_DWORD
Data: 0x1

Value 37
Name: E:\Informix\bin\xmppatch.exe
Type: REG_DWORD
Data: 0x1

Value 38
Name: E:\Informix\bin\xmpprof.exe
Type: REG_DWORD
Data: 0x1

Value 39
Name: E:\Informix\etc\arc_purge.sql
Type: REG_DWORD
Data: 0x1

Value 40

Name: E:\Informix\etc\BLDUTIL.BAT
Type: REG_DWORD
Data: 0x1

Value 41
Name: E:\Informix\etc\bldutil.in1
Type: REG_DWORD
Data: 0x1

Value 42
Name: E:\Informix\etc\bldutil.in2
Type: REG_DWORD
Data: 0x1

Value 43
Name: E:\Informix\etc\bldutil.in3
Type: REG_DWORD
Data: 0x1

Value 44
Name: E:\Informix\etc\bldutil.sh
Type: REG_DWORD
Data: 0x1

Value 45
Name: E:\Informix\etc\BUILDSMI.BAT
Type: REG_DWORD
Data: 0x1

Value 46
Name: E:\Informix\etc\buildsmi.in1
Type: REG_DWORD
Data: 0x1

Value 47
Name: E:\Informix\etc\buildsmi.in2
Type: REG_DWORD
Data: 0x1

Value 48
Name: E:\Informix\etc\buildsmi.in3
Type: REG_DWORD
Data: 0x1

Value 49
Name: E:\Informix\etc\buildsmi.in4
Type: REG_DWORD
Data: 0x1

Value 50
Name: E:\Informix\etc\CMDSHELL.ICO
Type: REG_DWORD
Data: 0x1

Value 51
Name: E:\Informix\etc\CNV50T60.SQL
Type: REG_DWORD
Data: 0x1

Value 52
Name: E:\Informix\etc\DBACCESS.ICO
Type: REG_DWORD
Data: 0x1

Value 53
Name: E:\Informix\etc\onconfig.std
Type: REG_DWORD
Data: 0x1

Value 54
Name: E:\Informix\etc\oninit.sym
Type: REG_DWORD
Data: 0x1

Value 55
Name: E:\Informix\etc\sym.out
Type: REG_DWORD
Data: 0x1

Value 56
Name: E:\Informix\etc\sysmaster.sql
Type: REG_DWORD
Data: 0x1

Value 57
Name: E:\Informix\etc\SYSUTILS.SQL
Type: REG_DWORD
Data: 0x1

Value 58
Name: E:\Informix\etc\XPG4_IS.SQL
Type: REG_DWORD
Data: 0x1

Value 59
Name: E:\Informix\help\ERRMESS.HLP
Type: REG_DWORD
Data: 0x1

Value 60
Name: E:\Informix\msg\4gl.iem
Type: REG_DWORD
Data: 0x1

Value 61
Name: E:\Informix\msg\4glusr.iem
Type: REG_DWORD
Data: 0x1

Value 62
Name: E:\Informix\msg\4glusr.msg
Type: REG_DWORD
Data: 0x1

Value 63
Name: E:\Informix\msg\all.iem
Type: REG_DWORD
Data: 0x1

Value 64
Name: E:\Informix\msg\archive.iem
Type: REG_DWORD
Data: 0x1

Value 65
Name: E:\Informix\msg\audit.iem
Type: REG_DWORD
Data: 0x1

Value 66
Name: E:\Informix\msg\be.iem
Type: REG_DWORD
Data: 0x1

Value 67
Name: E:\Informix\msg\c_err_e.dat
Type: REG_DWORD
Data: 0x1

Value 68
Name: E:\Informix\msg\c_err_f.dat
Type: REG_DWORD

Data: 0x1

Value 69
Name: E:\Informix\msg\dbacc.iem
Type: REG_DWORD
Data: 0x1

Value 70
Name: E:\Informix\msg\dbacc.lmk
Type: REG_DWORD
Data: 0x1

Value 71
Name: E:\Informix\msg\dbatool.iem
Type: REG_DWORD
Data: 0x1

Value 72
Name: E:\Informix\msg\dbised.iem
Type: REG_DWORD
Data: 0x1

Value 73
Name: E:\Informix\msg\dbised.lmk
Type: REG_DWORD
Data: 0x1

Value 74
Name: E:\Informix\msg\dbisedh.iem
Type: REG_DWORD
Data: 0x1

Value 75
Name: E:\Informix\msg\dbload.iem
Type: REG_DWORD
Data: 0x1

Value 76
Name: E:\Informix\msg\dbupd.iem
Type: REG_DWORD
Data: 0x1

Value 77
Name: E:\Informix\msg\ef77.iem
Type: REG_DWORD
Data: 0x1

Value 78
Name: E:\Informix\msg\english\itoxmsg.pam
Type: REG_DWORD
Data: 0x1

Value 79
Name: E:\Informix\msg\errmsg.txt
Type: REG_DWORD
Data: 0x1

Value 80
Name: E:\Informix\msg\errmsg_e.dat
Type: REG_DWORD
Data: 0x1

Value 81
Name: E:\Informix\msg\errmsg_f.dat
Type: REG_DWORD
Data: 0x1

Value 82
Name: E:\Informix\msg\esql.iem
Type: REG_DWORD
Data: 0x1

Value 83
Name: E:\Informix\msg\esqlc.iem
Type: REG_DWORD
Data: 0x1

Value 84
Name: E:\Informix\msg\esqlcob.iem
Type: REG_DWORD
Data: 0x1

Value 85
Name: E:\Informix\msg\fmt_e.dat
Type: REG_DWORD
Data: 0x1

Value 86
Name: E:\Informix\msg\fmt_f.dat
Type: REG_DWORD
Data: 0x1

Value 87
Name: E:\Informix\msg\formbld.iem
Type: REG_DWORD
Data: 0x1

Value 88
Name: E:\Informix\msg\forms.iem
Type: REG_DWORD
Data: 0x1

Value 89
Name: E:\Informix\msg\hlp_e.hpf
Type: REG_DWORD
Data: 0x1

Value 90
Name: E:\Informix\msg\hlp_f.hpf
Type: REG_DWORD
Data: 0x1

Value 91
Name: E:\Informix\msg\hlp_km_e.hpf
Type: REG_DWORD
Data: 0x1

Value 92
Name: E:\Informix\msg\hlp_km_f.hpf
Type: REG_DWORD
Data: 0x1

Value 93
Name: E:\Informix\msg\hlp_r_e.hpf
Type: REG_DWORD
Data: 0x1

Value 94
Name: E:\Informix\msg\hlp_r_f.hpf
Type: REG_DWORD
Data: 0x1

Value 95
Name: E:\Informix\msg\isam.iem
Type: REG_DWORD
Data: 0x1

Value 96
Name: E:\Informix\msg\itoxmsg.pam
Type: REG_DWORD
Data: 0x1

Value 97
Name: E:\Informix\msg\license.iem
Type: REG_DWORD
Data: 0x1

Value 98
Name: E:\Informix\msg\makefile
Type: REG_DWORD
Data: 0x1

Value 99
Name: E:\Informix\msg\menukey.iem
Type: REG_DWORD
Data: 0x1

Value 100
Name: E:\Informix\msg\mkem.iem
Type: REG_DWORD
Data: 0x1

Value 101
Name: E:\Informix\msg\mls.iem
Type: REG_DWORD
Data: 0x1

Value 102
Name: E:\Informix\msg\mls2.iem
Type: REG_DWORD
Data: 0x1

Value 103
Name: E:\Informix\msg\n4gl.iem
Type: REG_DWORD
Data: 0x1

Value 104
Name: E:\Informix\msg\n4glusr.iem
Type: REG_DWORD
Data: 0x1

Value 105
Name: E:\Informix\msg\necc.iem
Type: REG_DWORD
Data: 0x1

Value 106
Name: E:\Informix\msg\nerm.iem
Type: REG_DWORD
Data: 0x1

Value 107
Name: E:\Informix\msg\nesql.iem
Type: REG_DWORD
Data: 0x1

Value 108
Name: E:\Informix\msg\net.iem
Type: REG_DWORD
Data: 0x1

Value 109
Name: E:\Informix\msg\netsrv.iem
Type: REG_DWORD
Data: 0x1

Value 110
Name: E:\Informix\msg\nformbld.iem
Type: REG_DWORD
Data: 0x1

Value 111
Name: E:\Informix\msg\nforms.iem
Type: REG_DWORD
Data: 0x1

Value 112
Name: E:\Informix\msg\nls.iem
Type: REG_DWORD
Data: 0x1

Value 113
Name: E:\Informix\msg\ntol.iem
Type: REG_DWORD
Data: 0x1

Value 114
Name: E:\Informix\msg\onbar.iem
Type: REG_DWORD
Data: 0x1

Value 115
Name: E:\Informix\msg\oncheck.iem
Type: REG_DWORD
Data: 0x1

Value 116
Name: E:\Informix\msg\online.iem
Type: REG_DWORD
Data: 0x1

Value 117
Name: E:\Informix\msg\optical.iem
Type: REG_DWORD
Data: 0x1

Value 118
Name: E:\Informix\msg\os.iem
Type: REG_DWORD
Data: 0x1

Value 119
Name: E:\Informix\msg\pload.iem
Type: REG_DWORD
Data: 0x1

Value 120
Name: E:\Informix\msg\rds.iem
Type: REG_DWORD
Data: 0x1

Value 121
Name: E:\Informix\msg\rdsterm.iem
Type: REG_DWORD
Data: 0x1

Value 122
Name: E:\Informix\msg\rsam.iem
Type: REG_DWORD
Data: 0x1

Value 123
Name: E:\Informix\msg\secheck.iem
Type: REG_DWORD
Data: 0x1

Value 124
Name: E:\Informix\msg\security.iem
Type: REG_DWORD
Data: 0x1

Value 125
Name: E:\Informix\msg\shell.iem

Type: REG_DWORD
Data: 0x1

Value 126
Name: E:\Informix\msg\sql.iem
Type: REG_DWORD
Data: 0x1

Value 127
Name: E:\Informix\msg\sqli.iem
Type: REG_DWORD
Data: 0x1

Value 128
Name: E:\Informix\msg\util.iem
Type: REG_DWORD
Data: 0x1

Value 129
Name: E:\Informix\msg\xopen.iem
Type: REG_DWORD
Data: 0x1

Value 130
Name: E:\Informix\msg\xps.iem
Type: REG_DWORD
Data: 0x1

Value 131
Name: E:\Informix\release\README.WRI
Type: REG_DWORD
Data: 0x1

Key Name: SOFTWARE\Informix\Setup Framework\CurrentVersion\Setups\Groups
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:47 PM
Value 0
Name: \\INFORM1\Informix-Admin(Global)
Type: REG_DWORD
Data: 0x2

Key Name: SOFTWARE\Informix\Setup Framework\CurrentVersion\Setups\Program Manager Groups
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:49 PM
Value 0
Name: INFORMIX-XPS Server(Common)
Type: REG_DWORD
Data: 0x2

Key Name: SOFTWARE\Informix\Setup Framework\CurrentVersion\Setups\Registry Keys
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM
Value 0
Name: \\INFORM2\HKEY_LOCAL_MACHINE\SOFTWARE\Informix\RCE
Type: REG_DWORD
Data: 0x1

Value 1
Name: \\INFORM2\HKEY_LOCAL_MACHINE\SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Environment
Type: REG_DWORD
Data: 0x1

Value 2
Name: HKEY_LOCAL_MACHINE\SOFTWARE\Informix\SQLHOSTS
Type: REG_DWORD
Data: 0x1

Value 3
Name: HKEY_LOCAL_MACHINE\SOFTWARE\Informix\SQLHOSTS\ol_inform2
Type: REG_DWORD
Data: 0x1

Value 4
Name: HKEY_LOCAL_MACHINE\SOFTWARE\Informix\SQLHOSTS\ol_inform2\ol_inform2.1
Type: REG_DWORD
Data: 0x1

Value 5
Name: HKEY_LOCAL_MACHINE\SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security
Type: REG_DWORD
Data: 0x1

Value 6
Name: HKEY_LOCAL_MACHINE\SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security\IXAAO Group
Type: REG_DWORD
Data: 0x1

Value 7
Name: HKEY_LOCAL_MACHINE\SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security\IXDBSA Group
Type: REG_DWORD
Data: 0x1

Value 8
Name: HKEY_LOCAL_MACHINE\SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security\IXDBSSO Group
Type: REG_DWORD
Data: 0x1

Value 9
Name: HKEY_LOCAL_MACHINE\SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security\IXUSERS Group
Type: REG_DWORD
Data: 0x1

Value 10
Name: HKEY_LOCAL_MACHINE\SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security\Users
Type: REG_DWORD
Data: 0x1

Value 11
Name: HKEY_LOCAL_MACHINE\SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security\Users\Usernames
Type: REG_DWORD
Data: 0x1

Key Name: SOFTWARE\Informix\Setup Framework\CurrentVersion\Setups\Services
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM
Value 0
Name: \\INFORM2\XPSRCE
Type: REG_DWORD
Data: 0x1

Value 1
Name: MsgServ
Type: REG_DWORD
Data: 0x1

Key Name: SOFTWARE\Informix\Setup Framework\CurrentVersion\Setups\Shares
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM
Value 0
Name: \\INFORM2\SQEXPLN
Type: REG_DWORD
Data: 0x1

Key Name: SOFTWARE\Informix\Setup Framework\CurrentVersion\Setups\Users
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM
Value 0
Name: \\INFORM1\informix
Type: REG_DWORD
Data: 0x2

Key Name: SOFTWARE\Informix\SQLHOSTS
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM

Key Name: SOFTWARE\Informix\SQLHOSTS\ol_inform2
Class Name: <NO CLASS>
Last Write Time: 7/16/97 - 3:25 PM
Value 0
Name: HOST
Type: REG_SZ
Data: inform2

Value 1
Name: PROTOCOL
Type: REG_SZ
Data: olsoctcp

Value 2
Name: SERVICE
Type: REG_SZ
Data: turbo

Key Name: SOFTWARE\Informix\SQLHOSTS\ol_inform2\ol_inform2.1
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM
Value 0
Name: HOST
Type: REG_SZ
Data: inform2

Value 1
Name: OPTIONS
Type: REG_SZ
Data:

Value 2
Name: PROTOCOL
Type: REG_SZ
Data: olsoctcp

Value 3
Name: SERVICE
Type: REG_SZ
Data: turbo

Key Name: SOFTWARE\Informix\SQLHOSTS\ol_inform2.1
Class Name: Informix Registry Info
Last Write Time: 7/16/97 - 3:25 PM
Value 0
Name: HOST
Type: REG_SZ
Data: inform2

Value 1
Name: PROTOCOL
Type: REG_SZ
Data: olsoctcp

Value 2
Name: SERVICE
Type: REG_SZ
Data: turbo

Key Name: SOFTWARE\Informix\XPS8.00PG1
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:47 PM

Key Name: SOFTWARE\Informix\XPS8.00PG1\CurrentVersion
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:47 PM

Key Name: SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Environment
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM
Value 0
Name: COLLCHAR
Type: REG_SZ
Data: 1

Value 1
Name: DBNLS
Type: REG_SZ
Data: 0

Value 2
Name: INFORMIXDIR
Type: REG_SZ
Data: \\INFORM2\E$\Informix

Value 3
Name: INFORMIXSERVER
Type: REG_SZ
Data: ol_inform2.1

Value 4
Name: INFORMIXSQLHOSTS
Type: REG_SZ

Data: \\INFORM2

Value 5
Name: LANG
Type: REG_SZ
Data: English

Value 6
Name: ONCONFIG
Type: REG_SZ
Data: ONCONFIG

Value 7
Name: REGMACHINE
Type: REG_SZ
Data: \\INFORM2

Key Name:
SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM

Key Name:
SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security\IXAAO Group
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM
Value 0
Name: Name
Type: REG_SZ
Data: Informix-Admin

Key Name:
SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security\IXDBSA Group
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM
Value 0
Name: Domain
Type: REG_SZ
Data: LOTADISK

Value 1
Name: Name
Type: REG_SZ
Data: Informix-Admin

Key Name:
SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security\IXDBSSO Group
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM
Value 0
Name: Name
Type: REG_SZ
Data: Informix-Admin

Key Name:
SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security\IXUSERS Group
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM
Value 0
Name: Name
Type: REG_SZ
Data: *

Key Name:
SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security\Users
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:52 PM
Value 0
Name: UIDSeed
Type: REG_DWORD
Data: 0x6

Key Name:
SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security\Users\Usernames
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:52 PM
Value 0
Name: LOTADISK\informix
Type: REG_DWORD
Data: 0x6

Value 1
Name: UID6
Type: REG_SZ
Data: LOTADISK\informix

Key Name:
SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Setup
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM
Value 0
Name: InstallDate
Type: REG_SZ
Data: 07/08/1997, 15:49

Value 1
Name: Installed by
Type: REG_SZ
Data: LOTADISK\informix

Value 2
Name: PathName
Type: REG_SZ
Data: \\INFORM2\E$\Informix

Value 3
Name: Role Separation
Type: REG_DWORD
Data: 0

Value 4
Name: Server installed
Type: REG_DWORD
Data: 0x1

Value 5
Name: SoftwareType
Type: REG_SZ
Data: server

Value 6
Name: Version
Type: REG_SZ
Data: 8.00 PG1

Key Name:
SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Setup\Directories

```
Class Name:       <NO CLASS>
Last Write Time:  7/8/97 - 3:50 PM
Value 0
  Name:         \\INFORM2\E$\Informix
  Type:         REG_SZ
  Data:         New

Value 1
  Name:         \\INFORM2\E$\Informix\aaodir
  Type:         REG_SZ
  Data:         New

Value 2
  Name:         \\INFORM2\E$\Informix\bin
  Type:         REG_SZ
  Data:         New

Value 3
  Name:         \\INFORM2\E$\Informix\dbssodir
  Type:         REG_SZ
  Data:         New

Value 4
  Name:         \\INFORM2\E$\Informix\demo
  Type:         REG_SZ
  Data:         New

Value 5
  Name:         \\INFORM2\E$\Informix\demo\dbaccess
  Type:         REG_SZ
  Data:         New

Value 6
  Name:         \\INFORM2\E$\Informix\etc
  Type:         REG_SZ
  Data:         New

Value 7
  Name:         \\INFORM2\E$\Informix\forms
  Type:         REG_SZ
  Data:         New

Value 8
  Name:         \\INFORM2\E$\Informix\help
  Type:         REG_SZ
  Data:         New

Value 9
  Name:         \\INFORM2\E$\Informix\infxtmp
  Type:         REG_SZ
  Data:         New

Value 10
  Name:         \\INFORM2\E$\Informix\msg
  Type:         REG_SZ
  Data:         New

Value 11
  Name:         \\INFORM2\E$\Informix\release
  Type:         REG_SZ
  Data:         New

Value 12
  Name:         \\INFORM2\E$\Informix\sqexpln
  Type:         REG_SZ
  Data:         New

Value 13
  Name:         \\INFORM2\E$\tmp
  Type:         REG_SZ
  Data:         Old
```

```
Key Name:
SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Setup\Fil
es
Class Name:       <NO CLASS>
Last Write Time:  7/8/97 - 3:50 PM
Value 0
  Name:
\\INFORM2\\\INFORM2\admin$\system32\drivers\etc\hosts.eq
uiv
  Type:         REG_SZ
  Data:         Old

Value 1
  Name:         \\INFORM2\E$\Informix\aaodir\adtcfg.0
  Type:         REG_SZ
  Data:         New

Value 2
  Name:         \\INFORM2\E$\Informix\console.log
  Type:         REG_SZ
  Data:         New

Value 3
  Name:         \\INFORM2\E$\Informix\etc\.infos.ol_inform2
  Type:         REG_SZ
  Data:         New

Value 4
  Name:         \\INFORM2\E$\Informix\etc\bldutil.out
  Type:         REG_SZ
  Data:         New

Value 5
  Name:         \\INFORM2\E$\Informix\etc\buildsmi.out
  Type:         REG_SZ
  Data:         New

Value 6
  Name:
\\INFORM2\E$\Informix\etc\oncfg_ol_inform2.0
  Type:         REG_SZ
  Data:         New

Value 7
  Name:         \\INFORM2\E$\Informix\ETC\ONCONFIG
  Type:         REG_SZ
  Data:         New

Value 8
  Name:         \\INFORM2\E$\Informix\help\errmsg.ann
  Type:         REG_SZ
  Data:         New

Value 9
  Name:         \\INFORM2\E$\Informix\help\errmsg.ftg
  Type:         REG_SZ
  Data:         New

Value 10
  Name:         \\INFORM2\E$\Informix\help\errmsg.fts
  Type:         REG_SZ
  Data:         New

Value 11
  Name:         \\INFORM2\E$\Informix\help\errmsg.gid
  Type:         REG_SZ
  Data:         New

Value 12
```

Name: \\INFORM2\E$\Informix\help\errmsg.ph
Type: REG_SZ
Data: New

Value 13
Name: \\INFORM2\E$\Informix\online.log
Type: REG_SZ
Data: New

Value 14
Name: \\INFORM2\E$\Informix\SETENV.CMD
Type: REG_SZ
Data: New

Value 15
Name:
\\INFORM2\U$\IFMXDATA\ol_inform2\rootdbs1
Type: REG_SZ
Data: New

Value 16
Name: C:\TEMP\setup.log
Type: REG_SZ
Data: New

Value 17
Name: C:\WINNT\System32\online_service.log
Type: REG_SZ
Data: New

Value 18
Name: E:\Informix\bin\DBACCESS.EXE
Type: REG_SZ
Data: 1997-05-13

Value 19
Name: E:\Informix\bin\dbschema.exe
Type: REG_SZ
Data: 1997-04-04

Value 20
Name: E:\Informix\bin\dgtrans.ini
Type: REG_SZ
Data: 1996-05-14

Value 21
Name: E:\Informix\bin\dgtrans.sys
Type: REG_SZ
Data: 1996-05-28

Value 22
Name: E:\Informix\bin\ifmxdg.dll
Type: REG_SZ
Data: 1996-11-11

Value 23
Name: E:\Informix\bin\imacrouter.exe
Type: REG_SZ
Data: 1996-06-21

Value 24
Name: E:\Informix\bin\makedate.exe
Type: REG_SZ
Data: 1997-05-13

Value 25
Name: E:\Informix\bin\onevd.exe
Type: REG_SZ
Data: 1997-05-13

Value 26
Name: E:\Informix\bin\oninit.exe
Type: REG_SZ
Data: 1997-05-13

Value 27
Name: E:\Informix\bin\ONMODE.EXE
Type: REG_SZ
Data: 1997-05-13

Value 28
Name: E:\Informix\bin\ONPARAMS.EXE
Type: REG_SZ
Data: 1997-05-13

Value 29
Name: E:\Informix\bin\ONSPACES.EXE
Type: REG_SZ
Data: 1997-05-13

Value 30
Name: E:\Informix\bin\ONSTAT.EXE
Type: REG_SZ
Data: 1997-05-13

Value 31
Name: E:\Informix\bin\onutil.exe
Type: REG_SZ
Data: 1997-05-13

Value 32
Name: E:\Informix\bin\rce.exe
Type: REG_SZ
Data: 1996-11-11

Value 33
Name: E:\Informix\bin\rcmdsvc.exe
Type: REG_SZ
Data: 1996-12-02

Value 34
Name: E:\Informix\bin\readme.txt
Type: REG_SZ
Data: 1996-05-14

Value 35
Name: E:\Informix\bin\xboot.exe
Type: REG_SZ
Data: 1996-11-11

Value 36
Name: E:\Informix\bin\xctl.exe
Type: REG_SZ
Data: 1996-11-11

Value 37
Name: E:\Informix\bin\xmppatch.exe
Type: REG_SZ
Data: 1996-08-06

Value 38
Name: E:\Informix\bin\xmpprof.exe
Type: REG_SZ
Data: 1996-08-06

Value 39
Name: E:\Informix\etc\arc_purge.sql
Type: REG_SZ
Data: 1996-01-29

Value 40
Name: E:\Informix\etc\BLDUTIL.BAT

Type: REG_SZ
Data: 1996-05-02

Value 41
Name: E:\Informix\etc\bldutil.in1
Type: REG_SZ
Data: 1996-03-01

Value 42
Name: E:\Informix\etc\bldutil.in2
Type: REG_SZ
Data: 1996-03-01

Value 43
Name: E:\Informix\etc\bldutil.in3
Type: REG_SZ
Data: 1996-03-01

Value 44
Name: E:\Informix\etc\bldutil.sh
Type: REG_SZ
Data: 1996-03-01

Value 45
Name: E:\Informix\etc\BUILDSMI.BAT
Type: REG_SZ
Data: 1996-05-02

Value 46
Name: E:\Informix\etc\buildsmi.in1
Type: REG_SZ
Data: 1996-03-01

Value 47
Name: E:\Informix\etc\buildsmi.in2
Type: REG_SZ
Data: 1996-03-01

Value 48
Name: E:\Informix\etc\buildsmi.in3
Type: REG_SZ
Data: 1996-03-12

Value 49
Name: E:\Informix\etc\buildsmi.in4
Type: REG_SZ
Data: 1996-03-01

Value 50
Name: E:\Informix\etc\CMDSHELL.ICO
Type: REG_SZ
Data: 1996-05-09

Value 51
Name: E:\Informix\etc\CNV50T60.SQL
Type: REG_SZ
Data: 1994-05-20

Value 52
Name: E:\Informix\etc\DBACCESS.ICO
Type: REG_SZ
Data: 1996-05-09

Value 53
Name: E:\Informix\etc\onconfig.std
Type: REG_SZ
Data: 1997-05-14

Value 54
Name: E:\Informix\etc\oninit.sym
Type: REG_SZ
Data: 1997-04-22

Value 55
Name: E:\Informix\etc\sym.out
Type: REG_SZ
Data: 1996-11-11

Value 56
Name: E:\Informix\etc\sysmaster.sql
Type: REG_SZ
Data: 1996-06-07

Value 57
Name: E:\Informix\etc\SYSUTILS.SQL
Type: REG_SZ
Data: 1996-01-29

Value 58
Name: E:\Informix\etc\XPG4_IS.SQL
Type: REG_SZ
Data: 1994-09-06

Value 59
Name: E:\Informix\help\ERRMESS.HLP
Type: REG_SZ
Data: 1995-12-22

Value 60
Name: E:\Informix\msg\4gl.iem
Type: REG_SZ
Data: 1996-03-06

Value 61
Name: E:\Informix\msg\4glusr.iem
Type: REG_SZ
Data: 1996-03-06

Value 62
Name: E:\Informix\msg\4glusr.msg
Type: REG_SZ
Data: 1996-03-06

Value 63
Name: E:\Informix\msg\all.iem
Type: REG_SZ
Data: 1997-03-14

Value 64
Name: E:\Informix\msg\archive.iem
Type: REG_SZ
Data: 1997-03-14

Value 65
Name: E:\Informix\msg\audit.iem
Type: REG_SZ
Data: 1997-03-14

Value 66
Name: E:\Informix\msg\be.iem
Type: REG_SZ
Data: 1997-03-14

Value 67
Name: E:\Informix\msg\c_err_e.dat
Type: REG_SZ
Data: 1996-03-06

Value 68
Name: E:\Informix\msg\c_err_f.dat
Type: REG_SZ
Data: 1996-03-06

Value 69
Name: E:\Informix\msg\dbacc.iem
Type: REG_SZ
Data: 1997-03-14

Value 70
Name: E:\Informix\msg\dbacc.lmk
Type: REG_SZ
Data: 1996-08-19

Value 71
Name: E:\Informix\msg\dbatool.iem
Type: REG_SZ
Data: 1997-03-14

Value 72
Name: E:\Informix\msg\dbised.iem
Type: REG_SZ
Data: 1997-03-14

Value 73
Name: E:\Informix\msg\dbised.lmk
Type: REG_SZ
Data: 1996-08-19

Value 74
Name: E:\Informix\msg\dbisedh.iem
Type: REG_SZ
Data: 1997-03-14

Value 75
Name: E:\Informix\msg\dbload.iem
Type: REG_SZ
Data: 1997-03-14

Value 76
Name: E:\Informix\msg\dbupd.iem
Type: REG_SZ
Data: 1997-03-14

Value 77
Name: E:\Informix\msg\ef77.iem
Type: REG_SZ
Data: 1997-03-14

Value 78
Name: E:\Informix\msg\english\itoxmsg.pam
Type: REG_SZ
Data: 1996-08-19

Value 79
Name: E:\Informix\msg\errmsg.txt
Type: REG_SZ
Data: 1996-03-06

Value 80
Name: E:\Informix\msg\errmsg_e.dat
Type: REG_SZ
Data: 1996-03-06

Value 81
Name: E:\Informix\msg\errmsg_f.dat
Type: REG_SZ
Data: 1996-03-06

Value 82
Name: E:\Informix\msg\esql.iem
Type: REG_SZ
Data: 1996-01-22

Value 83
Name: E:\Informix\msg\esqlc.iem
Type: REG_SZ
Data: 1996-01-22

Value 84
Name: E:\Informix\msg\esqlcob.iem
Type: REG_SZ
Data: 1996-03-06

Value 85
Name: E:\Informix\msg\fmt_e.dat
Type: REG_SZ
Data: 1996-03-06

Value 86
Name: E:\Informix\msg\fmt_f.dat
Type: REG_SZ
Data: 1996-03-06

Value 87
Name: E:\Informix\msg\formbld.iem
Type: REG_SZ
Data: 1996-01-22

Value 88
Name: E:\Informix\msg\forms.iem
Type: REG_SZ
Data: 1996-01-22

Value 89
Name: E:\Informix\msg\hlp_e.hpf
Type: REG_SZ
Data: 1996-03-06

Value 90
Name: E:\Informix\msg\hlp_f.hpf
Type: REG_SZ
Data: 1996-03-06

Value 91
Name: E:\Informix\msg\hlp_km_e.hpf
Type: REG_SZ
Data: 1996-03-06

Value 92
Name: E:\Informix\msg\hlp_km_f.hpf
Type: REG_SZ
Data: 1996-03-06

Value 93
Name: E:\Informix\msg\hlp_r_e.hpf
Type: REG_SZ
Data: 1996-03-06

Value 94
Name: E:\Informix\msg\hlp_r_f.hpf
Type: REG_SZ
Data: 1996-03-06

Value 95
Name: E:\Informix\msg\isam.iem
Type: REG_SZ
Data: 1997-03-14

Value 96
Name: E:\Informix\msg\itoxmsg.pam
Type: REG_SZ
Data: 1996-08-19

Value 97

Name: E:\Informix\msg\license.iem
Type: REG_SZ
Data: 1997-03-14

Value 98
Name: E:\Informix\msg\makefile
Type: REG_SZ
Data: 1996-08-20

Value 99
Name: E:\Informix\msg\menukey.iem
Type: REG_SZ
Data: 1997-03-14

Value 100
Name: E:\Informix\msg\mkem.iem
Type: REG_SZ
Data: 1997-03-14

Value 101
Name: E:\Informix\msg\mls.iem
Type: REG_SZ
Data: 1996-01-04

Value 102
Name: E:\Informix\msg\mls2.iem
Type: REG_SZ
Data: 1996-01-04

Value 103
Name: E:\Informix\msg\n4gl.iem
Type: REG_SZ
Data: 1996-03-06

Value 104
Name: E:\Informix\msg\n4glusr.iem
Type: REG_SZ
Data: 1996-03-06

Value 105
Name: E:\Informix\msg\necc.iem
Type: REG_SZ
Data: 1996-01-22

Value 106
Name: E:\Informix\msg\nerm.iem
Type: REG_SZ
Data: 1997-03-14

Value 107
Name: E:\Informix\msg\nesql.iem
Type: REG_SZ
Data: 1996-01-22

Value 108
Name: E:\Informix\msg\net.iem
Type: REG_SZ
Data: 1997-03-14

Value 109
Name: E:\Informix\msg\netsrv.iem
Type: REG_SZ
Data: 1997-03-14

Value 110
Name: E:\Informix\msg\nformbld.iem
Type: REG_SZ
Data: 1996-01-22

Value 111
Name: E:\Informix\msg\nforms.iem
Type: REG_SZ
Data: 1996-01-22

Value 112
Name: E:\Informix\msg\nls.iem
Type: REG_SZ
Data: 1997-03-14

Value 113
Name: E:\Informix\msg\ntol.iem
Type: REG_SZ
Data: 1996-01-04

Value 114
Name: E:\Informix\msg\onbar.iem
Type: REG_SZ
Data: 1996-08-19

Value 115
Name: E:\Informix\msg\oncheck.iem
Type: REG_SZ
Data: 1997-03-14

Value 116
Name: E:\Informix\msg\online.iem
Type: REG_SZ
Data: 1997-03-14

Value 117
Name: E:\Informix\msg\optical.iem
Type: REG_SZ
Data: 1996-01-22

Value 118
Name: E:\Informix\msg\os.iem
Type: REG_SZ
Data: 1997-03-14

Value 119
Name: E:\Informix\msg\pload.iem
Type: REG_SZ
Data: 1997-03-14

Value 120
Name: E:\Informix\msg\rds.iem
Type: REG_SZ
Data: 1997-03-14

Value 121
Name: E:\Informix\msg\rdsterm.iem
Type: REG_SZ
Data: 1997-03-14

Value 122
Name: E:\Informix\msg\rsam.iem
Type: REG_SZ
Data: 1997-03-14

Value 123
Name: E:\Informix\msg\secheck.iem
Type: REG_SZ
Data: 1997-03-14

Value 124
Name: E:\Informix\msg\security.iem
Type: REG_SZ
Data: 1997-03-14

Value 125
Name: E:\Informix\msg\shell.iem
Type: REG_SZ

Data: 1997-03-14

Value 126
Name: E:\Informix\msg\sql.iem
Type: REG_SZ
Data: 1997-03-14

Value 127
Name: E:\Informix\msg\sqli.iem
Type: REG_SZ
Data: 1997-03-14

Value 128
Name: E:\Informix\msg\util.iem
Type: REG_SZ
Data: 1997-03-14

Value 129
Name: E:\Informix\msg\xopen.iem
Type: REG_SZ
Data: 1997-03-14

Value 130
Name: E:\Informix\msg\xps.iem
Type: REG_SZ
Data: 1997-03-14

Value 131
Name: E:\Informix\release\README.WRI
Type: REG_SZ
Data: 1996-01-30

Key Name:
SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Setup\Groups
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM

Key Name:
SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Setup\Groups\\|INFORM1|Informix-Admin(Global)
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM
Value 0
Name: Object
Type: REG_SZ
Data: Old

Key Name:
SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Setup\Program Manager Groups
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM

Key Name:
SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Setup\Program Manager Groups\INFORMIX-XPS Server(Common)
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM
Value 0
Name: Object
Type: REG_SZ
Data: Old

Key Name:
SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Setup\Program Manager Groups\INFORMIX-XPS Server(Common)\Icons
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM
Value 0
Name: Command Line Utilities
Type: REG_SZ
Data:

Value 1
Name: DBAccess
Type: REG_SZ
Data:

Value 2
Name: Find Error
Type: REG_SZ
Data:

Value 3
Name: OnMonitor
Type: REG_SZ
Data:

Value 4
Name: Release Notes
Type: REG_SZ
Data:

Value 5
Name: Uninstall
Type: REG_SZ
Data:

Key Name:
SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Setup\Registry Keys
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM
Value 0
Name:
\\INFORM2\HKEY_LOCAL_MACHINE\SOFTWARE\Informix\RCE
Type: REG_SZ
Data: New

Value 1
Name:
\\INFORM2\HKEY_LOCAL_MACHINE\SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Environment
Type: REG_SZ
Data: New

Value 2
Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Informix\SQLHOSTS
Type: REG_SZ
Data: New

Value 3
Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Informix\SQLHOSTS\ol_inform2
Type: REG_SZ
Data: New

Value 4
Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Informix\SQLHOSTS\ol_inform2\ol_inform2.1
Type: REG_SZ

Data: New

Value 5
Name: HKEY_LOCAL_MACHINE\SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security
Type: REG_SZ
Data: New

Value 6
Name: HKEY_LOCAL_MACHINE\SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security\IXAAO Group
Type: REG_SZ
Data: New

Value 7
Name: HKEY_LOCAL_MACHINE\SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security\IXDBSA Group
Type: REG_SZ
Data: New

Value 8
Name: HKEY_LOCAL_MACHINE\SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security\IXDBSSO Group
Type: REG_SZ
Data: New

Value 9
Name: HKEY_LOCAL_MACHINE\SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security\IXUSERS Group
Type: REG_SZ
Data: New

Value 10
Name: HKEY_LOCAL_MACHINE\SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security\Users
Type: REG_SZ
Data: New

Value 11
Name: HKEY_LOCAL_MACHINE\SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Security\Users\Usernames
Type: REG_SZ
Data: New

Key Name: SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Setup\Services
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM
Value 0
Name: \\INFORM2\XPSRCE
Type: REG_SZ
Data: New

Value 1
Name: MsgServ
Type: REG_SZ
Data: New

Key Name: SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Setup\Shares
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM
Value 0
Name: \\INFORM2\SQEXPLN
Type: REG_SZ
Data: New

Key Name: SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Setup\Users
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM

Key Name: SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Setup\Users\\|INFORM1|informix
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM
Value 0
Name: Object
Type: REG_SZ
Data: Old

Key Name: SOFTWARE\Informix\XPS8.00PG1\CurrentVersion\Setup\Users\\|INFORM1|informix\Container groups
Class Name: <NO CLASS>
Last Write Time: 7/8/97 - 3:50 PM
Value 0
Name: \\INFORM1\Administrators(Local)
Type: REG_SZ
Data:

Value 1
Name: \\INFORM1\Informix-Admin(Global)
Type: REG_SZ
Data:

Appendix B: Database Creation Statements

B-1: create_tpcd_database

```
@echo off
if "%1" == "" goto Usage
if "%DBDATE%" == "" goto EnvUsage

timer
echo create database
echo create database dssf%1 with log; | dbaccess
echo create the cogroup
onutil cr_group.onu

echo moving logs to another disk
call cr_lfiles.bat
call move_logs.bat


echo create lineitem slice for the line item table
onutil cr_line.onu

echo create order slice for the order tables
onutil cr_order.onu

echo create the cust slice for customer table
onutil cr_cust.onu
echo create supp slice for the supplier tables
onutil cr_supp.onu
echo create partsupp slice for the partsupp tables
onutil cr_partsupp.onu
echo create part slice for the parts tables
onutil cr_part.onu
echo create slice for ocod index
onutil cr_ocod.onu
echo create slice for lored index
onutil cr_lored.onu
echo create slice for psp index
onutil cr_psindex1.onu
echo create slice for pss index
onutil cr_psindex2.onu
echo create slice for orderkey index
onutil cr_okey.onu
echo create slice for lpqesod index
onutil cr_lpqesod.onu
echo create the temp slice for temporary table
onutil cr_temp.onu

echo "[START] creating tables ..."
sleep 5
timer
dbaccess dssf%1 create_tables.sql
sleep 5
timer
dbaccess dssf%1 load_tables.sql
timer
sleep 5
dbaccess dssf%1 create_indexes.sql
timer
sleep 5
dbaccess dssf%1 alter_it.sql
timer
sleep 5
dbaccess dssf%1 update_stats.sql
timer
xctl onmode -c
xctl onmode -kuy
echo "Database create and load complete !!"
echo "[END] TPC-D database ready ..."
timer
sleep 600

goto end

:Usage
echo Usage: create_tpcd_database dbsize
goto end

:EnvUsage
echo Environment Variable DBDATE is not set
goto end

:end
echo on
```

B-2: cr_group.onu

```
create cogroup ifmx from
ol_inform2.1;
```

B-3: move_logs.sh

```
echo moving logs to another disk
onutil cr_logslice.onu
xctl onmode -sy
sleep 30
echo onutil add_log.onu
onutil add_log.onu


echo alter cogroup ifmx reset backup;
echo alter cogroup ifmx reset backup; | onutil

xctl onmode -l
xctl onmode -l
xctl onmode -l
xctl onmode -l
xctl onmode -l
xctl onmode -c

echo onutil drop_log.onu
onutil drop_log.onu

xctl onmode -m
sleep 60
```

B-4: cr_line.onu

```
create dbslice l_month1 from cogroup ifmx chunk
"G:\30G\l_month1" size 340000;
create dbslice l_month2 from cogroup ifmx chunk
"H:\30G\l_month1" size 340000;
create dbslice l_month3 from cogroup ifmx chunk
"I:\30G\l_month1" size 340000;
create dbslice l_month4 from cogroup ifmx chunk
"J:\30G\l_month1" size 340000;
create dbslice l_month5 from cogroup ifmx chunk
"K:\30G\l_month1" size 340000;
create dbslice l_month6 from cogroup ifmx chunk
"L:\30G\l_month1" size 340000;
create dbslice l_month7 from cogroup ifmx chunk
"M:\30G\l_month1" size 340000;
create dbslice l_month8 from cogroup ifmx chunk
"n:\30G\l_month1" size 340000;
```

```
create dbslice l_month9 from cogroup ifmx chunk
"o:\30G\l_month1" size 340000;
create dbslice l_month10 from cogroup ifmx chunk
"p:\30G\l_month1" size 340000;
create dbslice l_month11 from cogroup ifmx chunk
"q:\30G\l_month1" size 340000;
create dbslice l_month12 from cogroup ifmx chunk
"r:\30G\l_month1" size 340000;
create dbslice l_month13 from cogroup ifmx chunk
"s:\30G\l_month1" size 340000;
create dbslice l_month14 from cogroup ifmx chunk
"t:\30G\l_month1" size 340000;
create dbslice l_month15 from cogroup ifmx chunk
"G:\30G\l_month2" size 340000;
create dbslice l_month16 from cogroup ifmx chunk
"H:\30G\l_month2" size 340000;
create dbslice l_month17 from cogroup ifmx chunk
"I:\30G\l_month2" size 340000;
create dbslice l_month18 from cogroup ifmx chunk
"J:\30G\l_month2" size 340000;
create dbslice l_month19 from cogroup ifmx chunk
"K:\30G\l_month2" size 340000;
create dbslice l_month20 from cogroup ifmx chunk
"L:\30G\l_month2" size 340000;
create dbslice l_month21 from cogroup ifmx chunk
"M:\30G\l_month2" size 340000;
create dbslice l_month22 from cogroup ifmx chunk
"n:\30G\l_month2" size 340000;
create dbslice l_month23 from cogroup ifmx chunk
"o:\30G\l_month2" size 340000;
create dbslice l_month24 from cogroup ifmx chunk
"p:\30G\l_month2" size 340000;
create dbslice l_month25 from cogroup ifmx chunk
"q:\30G\l_month2" size 340000;
create dbslice l_month26 from cogroup ifmx chunk
"r:\30G\l_month2" size 340000;
create dbslice l_month27 from cogroup ifmx chunk
"s:\30G\l_month2" size 340000;
create dbslice l_month28 from cogroup ifmx chunk
"t:\30G\l_month2" size 340000;
create dbslice l_month29 from cogroup ifmx chunk
"G:\30G\l_month3" size 340000;
create dbslice l_month30 from cogroup ifmx chunk
"H:\30G\l_month3" size 340000;
create dbslice l_month31 from cogroup ifmx chunk
"I:\30G\l_month3" size 340000;
create dbslice l_month32 from cogroup ifmx chunk
"J:\30G\l_month3" size 340000;
create dbslice l_month33 from cogroup ifmx chunk
"K:\30G\l_month3" size 340000;
create dbslice l_month34 from cogroup ifmx chunk
"L:\30G\l_month3" size 340000;
create dbslice l_month35 from cogroup ifmx chunk
"M:\30G\l_month3" size 340000;
create dbslice l_month36 from cogroup ifmx chunk
"n:\30G\l_month3" size 340000;
create dbslice l_month37 from cogroup ifmx chunk
"o:\30G\l_month3" size 340000;
create dbslice l_month38 from cogroup ifmx chunk
"p:\30G\l_month3" size 340000;
create dbslice l_month39 from cogroup ifmx chunk
"q:\30G\l_month3" size 340000;
create dbslice l_month40 from cogroup ifmx chunk
"r:\30G\l_month3" size 340000;
create dbslice l_month41 from cogroup ifmx chunk
"s:\30G\l_month3" size 340000;
create dbslice l_month42 from cogroup ifmx chunk
"t:\30G\l_month3" size 340000;
create dbslice l_month43 from cogroup ifmx chunk
"G:\30G\l_month4" size 340000;
create dbslice l_month44 from cogroup ifmx chunk
"H:\30G\l_month4" size 340000;
create dbslice l_month45 from cogroup ifmx chunk
"I:\30G\l_month4" size 340000;
create dbslice l_month46 from cogroup ifmx chunk
"J:\30G\l_month4" size 340000;
create dbslice l_month47 from cogroup ifmx chunk
"K:\30G\l_month4" size 340000;
create dbslice l_month48 from cogroup ifmx chunk
"L:\30G\l_month4" size 340000;
create dbslice l_month49 from cogroup ifmx chunk
"M:\30G\l_month4" size 340000;
create dbslice l_month50 from cogroup ifmx chunk
"n:\30G\l_month4" size 340000;
create dbslice l_month51 from cogroup ifmx chunk
"o:\30G\l_month4" size 340000;
create dbslice l_month52 from cogroup ifmx chunk
"p:\30G\l_month4" size 340000;
create dbslice l_month53 from cogroup ifmx chunk
"q:\30G\l_month4" size 340000;
create dbslice l_month54 from cogroup ifmx chunk
"r:\30G\l_month4" size 340000;
create dbslice l_month55 from cogroup ifmx chunk
"s:\30G\l_month4" size 340000;
create dbslice l_month56 from cogroup ifmx chunk
"t:\30G\l_month4" size 340000;
create dbslice l_month57 from cogroup ifmx chunk
"G:\30G\l_month5" size 340000;
create dbslice l_month58 from cogroup ifmx chunk
"H:\30G\l_month5" size 340000;
create dbslice l_month59 from cogroup ifmx chunk
"I:\30G\l_month5" size 340000;
create dbslice l_month60 from cogroup ifmx chunk
"J:\30G\l_month5" size 340000;
create dbslice l_month61 from cogroup ifmx chunk
"K:\30G\l_month5" size 340000;
create dbslice l_month62 from cogroup ifmx chunk
"L:\30G\l_month5" size 340000;
create dbslice l_month63 from cogroup ifmx chunk
"M:\30G\l_month5" size 340000;
create dbslice l_month64 from cogroup ifmx chunk
"n:\30G\l_month5" size 340000;
create dbslice l_month65 from cogroup ifmx chunk
"o:\30G\l_month5" size 340000;
create dbslice l_month66 from cogroup ifmx chunk
"p:\30G\l_month5" size 340000;
create dbslice l_month67 from cogroup ifmx chunk
"q:\30G\l_month5" size 340000;
create dbslice l_month68 from cogroup ifmx chunk
"r:\30G\l_month5" size 340000;
create dbslice l_month69 from cogroup ifmx chunk
"s:\30G\l_month5" size 340000;
create dbslice l_month70 from cogroup ifmx chunk
"t:\30G\l_month5" size 340000;
create dbslice l_month71 from cogroup ifmx chunk
"G:\30G\l_month6" size 340000;
create dbslice l_month72 from cogroup ifmx chunk
"H:\30G\l_month6" size 340000;
create dbslice l_month73 from cogroup ifmx chunk
"I:\30G\l_month6" size 340000;
create dbslice l_month74 from cogroup ifmx chunk
"J:\30G\l_month6" size 340000;
create dbslice l_month75 from cogroup ifmx chunk
"K:\30G\l_month6" size 340000;
create dbslice l_month76 from cogroup ifmx chunk
"L:\30G\l_month6" size 340000;
create dbslice l_month77 from cogroup ifmx chunk
"M:\30G\l_month6" size 340000;
create dbslice l_month78 from cogroup ifmx chunk
"n:\30G\l_month6" size 340000;
create dbslice l_month79 from cogroup ifmx chunk
"o:\30G\l_month6" size 340000;
```

```
create dbslice l_month80 from cogroup ifmx chunk
"p:\30G\l_month6" size 340000;
create dbslice l_month81 from cogroup ifmx chunk
"q:\30G\l_month6" size 340000;
create dbslice l_month82 from cogroup ifmx chunk
"r:\30G\l_month6" size 340000;
create dbslice l_month83 from cogroup ifmx chunk
"s:\30G\l_month6" size 340000;
create dbslice l_month84 from cogroup ifmx chunk
"t:\30G\l_month6" size 340000;
```

B-5: cr_order.onu

```
create dbslice o_month1 from cogroup ifmx chunk
"G:\30G\o_month1" size 85000;
create dbslice o_month2 from cogroup ifmx chunk
"H:\30G\o_month1" size 85000;
create dbslice o_month3 from cogroup ifmx chunk
"I:\30G\o_month1" size 85000;
create dbslice o_month4 from cogroup ifmx chunk
"J:\30G\o_month1" size 85000;
create dbslice o_month5 from cogroup ifmx chunk
"K:\30G\o_month1" size 85000;
create dbslice o_month6 from cogroup ifmx chunk
"L:\30G\o_month1" size 85000;
create dbslice o_month7 from cogroup ifmx chunk
"M:\30G\o_month1" size 85000;
create dbslice o_month8 from cogroup ifmx chunk
"n:\30G\o_month1" size 85000;
create dbslice o_month9 from cogroup ifmx chunk
"o:\30G\o_month1" size 85000;
create dbslice o_month10 from cogroup ifmx chunk
"p:\30G\o_month1" size 85000;
create dbslice o_month11 from cogroup ifmx chunk
"q:\30G\o_month1" size 85000;
create dbslice o_month12 from cogroup ifmx chunk
"r:\30G\o_month1" size 85000;
create dbslice o_month13 from cogroup ifmx chunk
"s:\30G\o_month1" size 85000;
create dbslice o_month14 from cogroup ifmx chunk
"t:\30G\o_month1" size 85000;
create dbslice o_month15 from cogroup ifmx chunk
"G:\30G\o_month2" size 85000;
create dbslice o_month16 from cogroup ifmx chunk
"H:\30G\o_month2" size 85000;
create dbslice o_month17 from cogroup ifmx chunk
"I:\30G\o_month2" size 85000;
create dbslice o_month18 from cogroup ifmx chunk
"J:\30G\o_month2" size 85000;
create dbslice o_month19 from cogroup ifmx chunk
"K:\30G\o_month2" size 85000;
create dbslice o_month20 from cogroup ifmx chunk
"L:\30G\o_month2" size 85000;
create dbslice o_month21 from cogroup ifmx chunk
"M:\30G\o_month2" size 85000;
create dbslice o_month22 from cogroup ifmx chunk
"n:\30G\o_month2" size 85000;
create dbslice o_month23 from cogroup ifmx chunk
"o:\30G\o_month2" size 85000;
create dbslice o_month24 from cogroup ifmx chunk
"p:\30G\o_month2" size 85000;
create dbslice o_month25 from cogroup ifmx chunk
"q:\30G\o_month2" size 85000;
create dbslice o_month26 from cogroup ifmx chunk
"r:\30G\o_month2" size 85000;
create dbslice o_month27 from cogroup ifmx chunk
"s:\30G\o_month2" size 85000;
create dbslice o_month28 from cogroup ifmx chunk
"t:\30G\o_month2" size 85000;
create dbslice o_month29 from cogroup ifmx chunk
"G:\30G\o_month3" size 85000;
create dbslice o_month30 from cogroup ifmx chunk
"H:\30G\o_month3" size 85000;
create dbslice o_month31 from cogroup ifmx chunk
"I:\30G\o_month3" size 85000;
create dbslice o_month32 from cogroup ifmx chunk
"J:\30G\o_month3" size 85000;
create dbslice o_month33 from cogroup ifmx chunk
"K:\30G\o_month3" size 85000;
create dbslice o_month34 from cogroup ifmx chunk
"L:\30G\o_month3" size 85000;
create dbslice o_month35 from cogroup ifmx chunk
"M:\30G\o_month3" size 85000;
create dbslice o_month36 from cogroup ifmx chunk
"n:\30G\o_month3" size 85000;
create dbslice o_month37 from cogroup ifmx chunk
"o:\30G\o_month3" size 85000;
create dbslice o_month38 from cogroup ifmx chunk
"p:\30G\o_month3" size 85000;
create dbslice o_month39 from cogroup ifmx chunk
"q:\30G\o_month3" size 85000;
create dbslice o_month40 from cogroup ifmx chunk
"r:\30G\o_month3" size 85000;
create dbslice o_month41 from cogroup ifmx chunk
"s:\30G\o_month3" size 85000;
create dbslice o_month42 from cogroup ifmx chunk
"t:\30G\o_month3" size 85000;
create dbslice o_month43 from cogroup ifmx chunk
"G:\30G\o_month4" size 85000;
create dbslice o_month44 from cogroup ifmx chunk
"H:\30G\o_month4" size 85000;
create dbslice o_month45 from cogroup ifmx chunk
"I:\30G\o_month4" size 85000;
create dbslice o_month46 from cogroup ifmx chunk
"J:\30G\o_month4" size 85000;
create dbslice o_month47 from cogroup ifmx chunk
"K:\30G\o_month4" size 85000;
create dbslice o_month48 from cogroup ifmx chunk
"L:\30G\o_month4" size 85000;
create dbslice o_month49 from cogroup ifmx chunk
"M:\30G\o_month4" size 85000;
create dbslice o_month50 from cogroup ifmx chunk
"n:\30G\o_month4" size 85000;
create dbslice o_month51 from cogroup ifmx chunk
"o:\30G\o_month4" size 85000;
create dbslice o_month52 from cogroup ifmx chunk
"p:\30G\o_month4" size 85000;
create dbslice o_month53 from cogroup ifmx chunk
"q:\30G\o_month4" size 85000;
create dbslice o_month54 from cogroup ifmx chunk
"r:\30G\o_month4" size 85000;
create dbslice o_month55 from cogroup ifmx chunk
"s:\30G\o_month4" size 85000;
create dbslice o_month56 from cogroup ifmx chunk
"t:\30G\o_month4" size 85000;
create dbslice o_month57 from cogroup ifmx chunk
"G:\30G\o_month5" size 85000;
create dbslice o_month58 from cogroup ifmx chunk
"H:\30G\o_month5" size 85000;
create dbslice o_month59 from cogroup ifmx chunk
"I:\30G\o_month5" size 85000;
create dbslice o_month60 from cogroup ifmx chunk
"J:\30G\o_month5" size 85000;
create dbslice o_month61 from cogroup ifmx chunk
"K:\30G\o_month5" size 85000;
create dbslice o_month62 from cogroup ifmx chunk
"L:\30G\o_month5" size 85000;
create dbslice o_month63 from cogroup ifmx chunk
"M:\30G\o_month5" size 85000;
create dbslice o_month64 from cogroup ifmx chunk
"n:\30G\o_month5" size 85000;
```

```
create dbslice o_month65 from cogroup ifmx chunk
"o:\30G\o_month5" size 85000;
create dbslice o_month66 from cogroup ifmx chunk
"p:\30G\o_month5" size 85000;
create dbslice o_month67 from cogroup ifmx chunk
"q:\30G\o_month5" size 85000;
create dbslice o_month68 from cogroup ifmx chunk
"r:\30G\o_month5" size 85000;
create dbslice o_month69 from cogroup ifmx chunk
"s:\30G\o_month5" size 85000;
create dbslice o_month70 from cogroup ifmx chunk
"t:\30G\o_month5" size 85000;
create dbslice o_month71 from cogroup ifmx chunk
"G:\30G\o_month6" size 85000;
create dbslice o_month72 from cogroup ifmx chunk
"H:\30G\o_month6" size 85000;
create dbslice o_month73 from cogroup ifmx chunk
"I:\30G\o_month6" size 85000;
create dbslice o_month74 from cogroup ifmx chunk
"J:\30G\o_month6" size 85000;
create dbslice o_month75 from cogroup ifmx chunk
"K:\30G\o_month6" size 85000;
create dbslice o_month76 from cogroup ifmx chunk
"L:\30G\o_month6" size 85000;
create dbslice o_month77 from cogroup ifmx chunk
"M:\30G\o_month6" size 85000;
create dbslice o_month78 from cogroup ifmx chunk
"n:\30G\o_month6" size 85000;
create dbslice o_month79 from cogroup ifmx chunk
"o:\30G\o_month6" size 85000;
create dbslice o_month80 from cogroup ifmx chunk
"p:\30G\o_month6" size 85000;
create dbslice o_month81 from cogroup ifmx chunk
"q:\30G\o_month6" size 85000;
create dbslice o_month82 from cogroup ifmx chunk
"r:\30G\o_month6" size 85000;
create dbslice o_month83 from cogroup ifmx chunk
"s:\30G\o_month6" size 85000;
create dbslice o_month84 from cogroup ifmx chunk
"t:\30G\o_month6" size 85000;
```

B-6: cr_cust.onu

```
create dbslice cust from cogroup ifmx chunk "G:\30G\c_data1"
size 40000,
cogroup ifmx chunk "H:\30G\c_data1" size 40000,
cogroup ifmx chunk "I:\30G\c_data1" size 40000,
cogroup ifmx chunk "J:\30G\c_data1" size 40000,
cogroup ifmx chunk "K:\30G\c_data1" size 40000,
cogroup ifmx chunk "L:\30G\c_data1" size 40000,
cogroup ifmx chunk "M:\30G\c_data1" size 40000,
cogroup ifmx chunk "n:\30G\c_data1" size 40000,
cogroup ifmx chunk "o:\30G\c_data1" size 40000,
cogroup ifmx chunk "p:\30G\c_data1" size 40000,
cogroup ifmx chunk "q:\30G\c_data1" size 40000,
cogroup ifmx chunk "r:\30G\c_data1" size 40000,
cogroup ifmx chunk "s:\30G\c_data1" size 40000,
cogroup ifmx chunk "t:\30G\c_data1" size 40000,

cogroup ifmx chunk "G:\30G\c_data2" size 40000,
cogroup ifmx chunk "H:\30G\c_data2" size 40000,
cogroup ifmx chunk "I:\30G\c_data2" size 40000,
cogroup ifmx chunk "J:\30G\c_data2" size 40000,
cogroup ifmx chunk "K:\30G\c_data2" size 40000,
cogroup ifmx chunk "L:\30G\c_data2" size 40000,
cogroup ifmx chunk "M:\30G\c_data2" size 40000,
cogroup ifmx chunk "n:\30G\c_data2" size 40000,
cogroup ifmx chunk "o:\30G\c_data2" size 40000,
cogroup ifmx chunk "p:\30G\c_data2" size 40000,
cogroup ifmx chunk "q:\30G\c_data2" size 40000,
cogroup ifmx chunk "r:\30G\c_data2" size 40000,
cogroup ifmx chunk "s:\30G\c_data2" size 40000,
cogroup ifmx chunk "t:\30G\c_data2" size 40000;
```

B-7: cr_supp.onu

```
create dbslice supp from cogroup ifmx chunk
"G:\30G\s_data1" size 10600,
cogroup ifmx chunk "H:\30G\s_data1" size 10600,
cogroup ifmx chunk "I:\30G\s_data1" size 10600,
cogroup ifmx chunk "J:\30G\s_data1" size 10600,
cogroup ifmx chunk "K:\30G\s_data1" size 10600,
cogroup ifmx chunk "L:\30G\s_data1" size 10600,
cogroup ifmx chunk "M:\30G\s_data1" size 10600,
cogroup ifmx chunk "n:\30G\s_data1" size 10600,
cogroup ifmx chunk "o:\30G\s_data1" size 10600,
cogroup ifmx chunk "p:\30G\s_data1" size 10600,
cogroup ifmx chunk "q:\30G\s_data1" size 10600,
cogroup ifmx chunk "r:\30G\s_data1" size 10600,
cogroup ifmx chunk "s:\30G\s_data1" size 10600,
cogroup ifmx chunk "t:\30G\s_data1" size 10600,
cogroup ifmx chunk "G:\30G\s_data2" size 10600,
cogroup ifmx chunk "H:\30G\s_data2" size 10600,
cogroup ifmx chunk "I:\30G\s_data2" size 10600,
cogroup ifmx chunk "J:\30G\s_data2" size 10600,
cogroup ifmx chunk "K:\30G\s_data2" size 10600,
cogroup ifmx chunk "L:\30G\s_data2" size 10600,
cogroup ifmx chunk "M:\30G\s_data2" size 10600,
cogroup ifmx chunk "n:\30G\s_data2" size 10600,
cogroup ifmx chunk "o:\30G\s_data2" size 10600,
cogroup ifmx chunk "p:\30G\s_data2" size 10600,
cogroup ifmx chunk "q:\30G\s_data2" size 10600,
cogroup ifmx chunk "r:\30G\s_data2" size 10600,
cogroup ifmx chunk "s:\30G\s_data2" size 10600,
cogroup ifmx chunk "t:\30G\s_data2" size 10600;
```

B-8: cr_part.onu

```
create dbslice part from cogroup ifmx chunk "G:\30G\p_data1"
size 40000,
cogroup ifmx chunk "H:\30G\p_data1" size 40000,
cogroup ifmx chunk "I:\30G\p_data1" size 40000,
cogroup ifmx chunk "J:\30G\p_data1" size 40000,
cogroup ifmx chunk "K:\30G\p_data1" size 40000,
cogroup ifmx chunk "L:\30G\p_data1" size 40000,
cogroup ifmx chunk "M:\30G\p_data1" size 40000,
cogroup ifmx chunk "n:\30G\p_data1" size 40000,
cogroup ifmx chunk "o:\30G\p_data1" size 40000,
cogroup ifmx chunk "p:\30G\p_data1" size 40000,
cogroup ifmx chunk "q:\30G\p_data1" size 40000,
cogroup ifmx chunk "r:\30G\p_data1" size 40000,
cogroup ifmx chunk "s:\30G\p_data1" size 40000,
cogroup ifmx chunk "t:\30G\p_data1" size 40000,
cogroup ifmx chunk "G:\30G\p_data2" size 40000,
cogroup ifmx chunk "H:\30G\p_data2" size 40000,
cogroup ifmx chunk "I:\30G\p_data2" size 40000,
cogroup ifmx chunk "J:\30G\p_data2" size 40000,
cogroup ifmx chunk "K:\30G\p_data2" size 40000,
cogroup ifmx chunk "L:\30G\p_data2" size 40000,
cogroup ifmx chunk "M:\30G\p_data2" size 40000,
cogroup ifmx chunk "n:\30G\p_data2" size 40000,
cogroup ifmx chunk "o:\30G\p_data2" size 40000,
cogroup ifmx chunk "p:\30G\p_data2" size 40000,
cogroup ifmx chunk "q:\30G\p_data2" size 40000,
cogroup ifmx chunk "r:\30G\p_data2" size 40000,
cogroup ifmx chunk "s:\30G\p_data2" size 40000,
cogroup ifmx chunk "t:\30G\p_data2" size 40000;
```

B-9: cr_partsupp.onu

```
create dbslice ps_supp1 from cogroup ifmx chunk
"G:\30G\ps_supp1" size 200000,
cogroup ifmx chunk "H:\30G\ps_supp1" size 200000,
cogroup ifmx chunk "I:\30G\ps_supp1" size 200000,
cogroup ifmx chunk "J:\30G\ps_supp1" size 200000,
cogroup ifmx chunk "K:\30G\ps_supp1" size 200000,
cogroup ifmx chunk "L:\30G\ps_supp1" size 200000,
cogroup ifmx chunk "M:\30G\ps_supp1" size 200000,
cogroup ifmx chunk "n:\30G\ps_supp1" size 200000,
cogroup ifmx chunk "o:\30G\ps_supp1" size 200000,
cogroup ifmx chunk "p:\30G\ps_supp1" size 200000,
cogroup ifmx chunk "q:\30G\ps_supp1" size 200000,
cogroup ifmx chunk "r:\30G\ps_supp1" size 200000,
cogroup ifmx chunk "s:\30G\ps_supp1" size 200000,
cogroup ifmx chunk "t:\30G\ps_supp1" size 200000,
cogroup ifmx chunk "G:\30G\ps_supp2" size 200000,
cogroup ifmx chunk "H:\30G\ps_supp2" size 200000,
cogroup ifmx chunk "I:\30G\ps_supp2" size 200000,
cogroup ifmx chunk "J:\30G\ps_supp2" size 200000,
cogroup ifmx chunk "K:\30G\ps_supp2" size 200000,
cogroup ifmx chunk "L:\30G\ps_supp2" size 200000,
cogroup ifmx chunk "M:\30G\ps_supp2" size 200000,
cogroup ifmx chunk "n:\30G\ps_supp2" size 200000,
cogroup ifmx chunk "o:\30G\ps_supp2" size 200000,
cogroup ifmx chunk "p:\30G\ps_supp2" size 200000,
cogroup ifmx chunk "q:\30G\ps_supp2" size 200000,
cogroup ifmx chunk "r:\30G\ps_supp2" size 200000,
cogroup ifmx chunk "s:\30G\ps_supp2" size 200000,
cogroup ifmx chunk "t:\30G\ps_supp2" size 200000;
```

B-10: cr_temp.onu

```
create temp dbslice temp_slice1 from cogroup ifmx chunk
"G:\30G\temp1" size 1024000;

create temp dbslice temp_slice2 from cogroup ifmx chunk
"H:\30G\temp1" size 1024000;

create temp dbslice temp_slice3 from cogroup ifmx chunk
"I:\30G\temp1" size 1024000;

create temp dbslice temp_slice4 from cogroup ifmx chunk
"J:\30G\temp1" size 1024000;

create temp dbslice temp_slice5 from cogroup ifmx chunk
"K:\30G\temp1" size 1024000;

create temp dbslice temp_slice6 from cogroup ifmx chunk
"L:\30G\temp1" size 1024000;

create temp dbslice temp_slice7 from cogroup ifmx chunk
"M:\30G\temp1" size 1024000;


create temp dbslice temp_slice11 from cogroup ifmx chunk
"n:\30G\temp1" size 1024000;

create temp dbslice temp_slice12 from cogroup ifmx chunk
"o:\30G\temp1" size 1024000;

create temp dbslice temp_slice13 from cogroup ifmx chunk
"p:\30G\temp1" size 1024000;

create temp dbslice temp_slice14 from cogroup ifmx chunk
"q:\30G\temp1" size 1024000;

create temp dbslice temp_slice15 from cogroup ifmx chunk
"r:\30G\temp1" size 1024000;

create temp dbslice temp_slice16 from cogroup ifmx chunk
"s:\30G\temp1" size 1024000;

create temp dbslice temp_slice17 from cogroup ifmx chunk
"t:\30G\temp1" size 1024000;
```

B-11: cr_ocod.onu

```
create dbslice o_ocod1 from cogroup ifmx chunk
"G:\30G\o_ocod1" size 80000;
create dbslice o_ocod2 from cogroup ifmx chunk
"H:\30G\o_ocod1" size 80000;
create dbslice o_ocod3 from cogroup ifmx chunk
"I:\30G\o_ocod1" size 80000;
create dbslice o_ocod4 from cogroup ifmx chunk
"J:\30G\o_ocod1" size 80000;
create dbslice o_ocod5 from cogroup ifmx chunk
"K:\30G\o_ocod1" size 80000;
create dbslice o_ocod6 from cogroup ifmx chunk
"L:\30G\o_ocod1" size 80000;
create dbslice o_ocod7 from cogroup ifmx chunk
"M:\30G\o_ocod1" size 80000;
create dbslice o_ocod8 from cogroup ifmx chunk
"n:\30G\o_ocod1" size 80000;
create dbslice o_ocod9 from cogroup ifmx chunk
"o:\30G\o_ocod1" size 80000;
create dbslice o_ocod10 from cogroup ifmx chunk
"p:\30G\o_ocod1" size 80000;
create dbslice o_ocod11 from cogroup ifmx chunk
"q:\30G\o_ocod1" size 80000;
create dbslice o_ocod12 from cogroup ifmx chunk
"r:\30G\o_ocod1" size 80000;
create dbslice o_ocod13 from cogroup ifmx chunk
"s:\30G\o_ocod1" size 80000;
create dbslice o_ocod14 from cogroup ifmx chunk
"t:\30G\o_ocod1" size 80000;
create dbslice o_ocod15 from cogroup ifmx chunk
"G:\30G\o_ocod2" size 80000;
create dbslice o_ocod16 from cogroup ifmx chunk
"H:\30G\o_ocod2" size 80000;
create dbslice o_ocod17 from cogroup ifmx chunk
"I:\30G\o_ocod2" size 80000;
create dbslice o_ocod18 from cogroup ifmx chunk
"J:\30G\o_ocod2" size 80000;
create dbslice o_ocod19 from cogroup ifmx chunk
"K:\30G\o_ocod2" size 80000;
create dbslice o_ocod20 from cogroup ifmx chunk
"L:\30G\o_ocod2" size 80000;
create dbslice o_ocod21 from cogroup ifmx chunk
"M:\30G\o_ocod2" size 80000;
create dbslice o_ocod22 from cogroup ifmx chunk
"n:\30G\o_ocod2" size 80000;
create dbslice o_ocod23 from cogroup ifmx chunk
"o:\30G\o_ocod2" size 80000;
create dbslice o_ocod24 from cogroup ifmx chunk
"p:\30G\o_ocod2" size 80000;
create dbslice o_ocod25 from cogroup ifmx chunk
"q:\30G\o_ocod2" size 80000;
create dbslice o_ocod26 from cogroup ifmx chunk
"r:\30G\o_ocod2" size 80000;
create dbslice o_ocod27 from cogroup ifmx chunk
"s:\30G\o_ocod2" size 80000;
create dbslice o_ocod28 from cogroup ifmx chunk
"t:\30G\o_ocod2" size 80000;
```

B-12: cr_lored.onu

```
create dbslice l_lored from cogroup ifmx chunk
"G:\30G\l_ored1" size 340000,
cogroup ifmx chunk "H:\30G\l_ored1" size 340000,
cogroup ifmx chunk "I:\30G\l_ored1" size 340000,
cogroup ifmx chunk "J:\30G\l_ored1" size 340000,
cogroup ifmx chunk "K:\30G\l_ored1" size 340000,
cogroup ifmx chunk "L:\30G\l_ored1" size 340000,
cogroup ifmx chunk "M:\30G\l_ored1" size 340000,
cogroup ifmx chunk "n:\30G\l_ored1" size 340000,
cogroup ifmx chunk "o:\30G\l_ored1" size 340000,
cogroup ifmx chunk "p:\30G\l_ored1" size 340000,
cogroup ifmx chunk "q:\30G\l_ored1" size 340000,
cogroup ifmx chunk "r:\30G\l_ored1" size 340000,
cogroup ifmx chunk "s:\30G\l_ored1" size 340000,
cogroup ifmx chunk "t:\30G\l_ored1" size 340000,

cogroup ifmx chunk "G:\30G\l_ored2" size 340000,
cogroup ifmx chunk "H:\30G\l_ored2" size 340000,
cogroup ifmx chunk "I:\30G\l_ored2" size 340000,
cogroup ifmx chunk "J:\30G\l_ored2" size 340000,
cogroup ifmx chunk "K:\30G\l_ored2" size 340000,
cogroup ifmx chunk "L:\30G\l_ored2" size 340000,
cogroup ifmx chunk "M:\30G\l_ored2" size 340000,
cogroup ifmx chunk "n:\30G\l_ored2" size 340000,
cogroup ifmx chunk "o:\30G\l_ored2" size 340000,
cogroup ifmx chunk "p:\30G\l_ored2" size 340000,
cogroup ifmx chunk "q:\30G\l_ored2" size 340000,
cogroup ifmx chunk "r:\30G\l_ored2" size 340000,
cogroup ifmx chunk "s:\30G\l_ored2" size 340000,
cogroup ifmx chunk "t:\30G\l_ored2" size 340000;
```

B-13: cr_psindex1.onu

```
create dbslice ps_pspss1 from cogroup ifmx chunk
"G:\30G\ps_pspss1" size 34000;
create dbslice ps_pspss2 from cogroup ifmx chunk
"H:\30G\ps_pspss1" size 34000;
create dbslice ps_pspss3 from cogroup ifmx chunk
"I:\30G\ps_pspss1" size 34000;
create dbslice ps_pspss4 from cogroup ifmx chunk
"J:\30G\ps_pspss1" size 34000;
create dbslice ps_pspss5 from cogroup ifmx chunk
"K:\30G\ps_pspss1" size 34000;
create dbslice ps_pspss6 from cogroup ifmx chunk
"L:\30G\ps_pspss1" size 34000;
create dbslice ps_pspss7 from cogroup ifmx chunk
"M:\30G\ps_pspss1" size 34000;
create dbslice ps_pspss8 from cogroup ifmx chunk
"n:\30G\ps_pspss1" size 34000;
create dbslice ps_pspss9 from cogroup ifmx chunk
"o:\30G\ps_pspss1" size 34000;
create dbslice ps_pspss10 from cogroup ifmx chunk
"p:\30G\ps_pspss1" size 34000;
create dbslice ps_pspss11 from cogroup ifmx chunk
"q:\30G\ps_pspss1" size 34000;
create dbslice ps_pspss12 from cogroup ifmx chunk
"r:\30G\ps_pspss1" size 34000;
create dbslice ps_pspss13 from cogroup ifmx chunk
"s:\30G\ps_pspss1" size 34000;
create dbslice ps_pspss14 from cogroup ifmx chunk
"t:\30G\ps_pspss1" size 34000;
create dbslice ps_pspss15 from cogroup ifmx chunk
"G:\30G\ps_pspss2" size 34000;
create dbslice ps_pspss16 from cogroup ifmx chunk
"H:\30G\ps_pspss2" size 34000;
create dbslice ps_pspss17 from cogroup ifmx chunk
"I:\30G\ps_pspss2" size 34000;
create dbslice ps_pspss18 from cogroup ifmx chunk
"J:\30G\ps_pspss2" size 34000;
create dbslice ps_pspss19 from cogroup ifmx chunk
"K:\30G\ps_pspss2" size 34000;
create dbslice ps_pspss20 from cogroup ifmx chunk
"L:\30G\ps_pspss2" size 34000;
create dbslice ps_pspss21 from cogroup ifmx chunk
"M:\30G\ps_pspss2" size 34000;
create dbslice ps_pspss22 from cogroup ifmx chunk
"n:\30G\ps_pspss2" size 34000;
create dbslice ps_pspss23 from cogroup ifmx chunk
"o:\30G\ps_pspss2" size 34000;
create dbslice ps_pspss24 from cogroup ifmx chunk
"p:\30G\ps_pspss2" size 34000;
create dbslice ps_pspss25 from cogroup ifmx chunk
"q:\30G\ps_pspss2" size 34000;
create dbslice ps_pspss26 from cogroup ifmx chunk
"r:\30G\ps_pspss2" size 34000;
create dbslice ps_pspss27 from cogroup ifmx chunk
"s:\30G\ps_pspss2" size 34000;
create dbslice ps_pspss28 from cogroup ifmx chunk
"t:\30G\ps_pspss2" size 34000;
```

B-14: cr_psindex2.onu

```
create dbslice ps_spsa1 from cogroup ifmx chunk
"G:\30G\ps_spsa1" size 38000;
create dbslice ps_spsa2 from cogroup ifmx chunk
"H:\30G\ps_spsa1" size 38000;
create dbslice ps_spsa3 from cogroup ifmx chunk
"I:\30G\ps_spsa1" size 38000;
create dbslice ps_spsa4 from cogroup ifmx chunk
"J:\30G\ps_spsa1" size 38000;
create dbslice ps_spsa5 from cogroup ifmx chunk
"K:\30G\ps_spsa1" size 38000;
create dbslice ps_spsa6 from cogroup ifmx chunk
"L:\30G\ps_spsa1" size 38000;
create dbslice ps_spsa7 from cogroup ifmx chunk
"M:\30G\ps_spsa1" size 38000;
create dbslice ps_spsa8 from cogroup ifmx chunk
"n:\30G\ps_spsa1" size 38000;
create dbslice ps_spsa9 from cogroup ifmx chunk
"o:\30G\ps_spsa1" size 38000;
create dbslice ps_spsa10 from cogroup ifmx chunk
"p:\30G\ps_spsa1" size 38000;
create dbslice ps_spsa11 from cogroup ifmx chunk
"q:\30G\ps_spsa1" size 38000;
create dbslice ps_spsa12 from cogroup ifmx chunk
"r:\30G\ps_spsa1" size 38000;
create dbslice ps_spsa13 from cogroup ifmx chunk
"s:\30G\ps_spsa1" size 38000;
create dbslice ps_spsa14 from cogroup ifmx chunk
"t:\30G\ps_spsa1" size 38000;
create dbslice ps_spsa15 from cogroup ifmx chunk
"G:\30G\ps_spsa2" size 38000;
create dbslice ps_spsa16 from cogroup ifmx chunk
"H:\30G\ps_spsa2" size 38000;
create dbslice ps_spsa17 from cogroup ifmx chunk
"I:\30G\ps_spsa2" size 38000;
create dbslice ps_spsa18 from cogroup ifmx chunk
"J:\30G\ps_spsa2" size 38000;
create dbslice ps_spsa19 from cogroup ifmx chunk
"K:\30G\ps_spsa2" size 38000;
create dbslice ps_spsa20 from cogroup ifmx chunk
"L:\30G\ps_spsa2" size 38000;
create dbslice ps_spsa21 from cogroup ifmx chunk
"M:\30G\ps_spsa2" size 38000;
create dbslice ps_spsa22 from cogroup ifmx chunk
"n:\30G\ps_spsa2" size 38000;
create dbslice ps_spsa23 from cogroup ifmx chunk
"o:\30G\ps_spsa2" size 38000;
```

```
create dbslice ps_spsa24 from cogroup ifmx chunk
"p:\30G\ps_spsa2" size 38000;
create dbslice ps_spsa25 from cogroup ifmx chunk
"q:\30G\ps_spsa2" size 38000;
create dbslice ps_spsa26 from cogroup ifmx chunk
"r:\30G\ps_spsa2" size 38000;
create dbslice ps_spsa27 from cogroup ifmx chunk
"s:\30G\ps_spsa2" size 38000;
create dbslice ps_spsa28 from cogroup ifmx chunk
"t:\30G\ps_spsa2" size 38000;
```

B-15: cr_okey.onu

```
create dbslice o_okey1 from cogroup ifmx chunk
"G:\30G\o_key1" size 42000,
cogroup ifmx chunk "H:\30G\o_key1" size 42000,
cogroup ifmx chunk "I:\30G\o_key1" size 42000,
cogroup ifmx chunk "J:\30G\o_key1" size 42000,
cogroup ifmx chunk "K:\30G\o_key1" size 42000,
cogroup ifmx chunk "L:\30G\o_key1" size 42000,
cogroup ifmx chunk "M:\30G\o_key1" size 42000,
cogroup ifmx chunk "n:\30G\o_key1" size 42000,
cogroup ifmx chunk "o:\30G\o_key1" size 42000,
cogroup ifmx chunk "p:\30G\o_key1" size 42000,
cogroup ifmx chunk "q:\30G\o_key1" size 42000,
cogroup ifmx chunk "r:\30G\o_key1" size 42000,
cogroup ifmx chunk "s:\30G\o_key1" size 42000,
cogroup ifmx chunk "t:\30G\o_key1" size 42000,
cogroup ifmx chunk "G:\30G\o_key2" size 42000,
cogroup ifmx chunk "H:\30G\o_key2" size 42000,
cogroup ifmx chunk "I:\30G\o_key2" size 42000,
cogroup ifmx chunk "J:\30G\o_key2" size 42000,
cogroup ifmx chunk "K:\30G\o_key2" size 42000,
cogroup ifmx chunk "L:\30G\o_key2" size 42000,
cogroup ifmx chunk "M:\30G\o_key2" size 42000,
cogroup ifmx chunk "n:\30G\o_key2" size 42000,
cogroup ifmx chunk "o:\30G\o_key2" size 42000,
cogroup ifmx chunk "p:\30G\o_key2" size 42000,
cogroup ifmx chunk "q:\30G\o_key2" size 42000,
cogroup ifmx chunk "r:\30G\o_key2" size 42000,
cogroup ifmx chunk "s:\30G\o_key2" size 42000,
cogroup ifmx chunk "t:\30G\o_key2" size 42000;
```

B-16: cr_lpqesod.onu

```
create dbslice l_pqesod1 from cogroup ifmx chunk
"G:\30g\l_pqesod1" size 231000;
create dbslice l_pqesod2 from cogroup ifmx chunk
"H:\30g\l_pqesod1" size 231000;
create dbslice l_pqesod3 from cogroup ifmx chunk
"I:\30g\l_pqesod1" size 231000;
create dbslice l_pqesod4 from cogroup ifmx chunk
"J:\30g\l_pqesod1" size 231000;
create dbslice l_pqesod5 from cogroup ifmx chunk
"K:\30g\l_pqesod1" size 231000;
create dbslice l_pqesod6 from cogroup ifmx chunk
"L:\30g\l_pqesod1" size 231000;
create dbslice l_pqesod7 from cogroup ifmx chunk
"M:\30g\l_pqesod1" size 231000;
create dbslice l_pqesod8 from cogroup ifmx chunk
"n:\30g\l_pqesod1" size 231000;
create dbslice l_pqesod9 from cogroup ifmx chunk
"o:\30g\l_pqesod1" size 231000;
create dbslice l_pqesod10 from cogroup ifmx chunk
"p:\30g\l_pqesod1" size 231000;
create dbslice l_pqesod11 from cogroup ifmx chunk
"q:\30g\l_pqesod1" size 231000;
create dbslice l_pqesod12 from cogroup ifmx chunk
"r:\30g\l_pqesod1" size 231000;
create dbslice l_pqesod13 from cogroup ifmx chunk
"s:\30g\l_pqesod1" size 231000;
create dbslice l_pqesod14 from cogroup ifmx chunk
"t:\30g\l_pqesod1" size 231000;
create dbslice l_pqesod15 from cogroup ifmx chunk
"G:\30g\l_pqesod2" size 231000;
create dbslice l_pqesod16 from cogroup ifmx chunk
"H:\30g\l_pqesod2" size 231000;
create dbslice l_pqesod17 from cogroup ifmx chunk
"I:\30g\l_pqesod2" size 231000;
create dbslice l_pqesod18 from cogroup ifmx chunk
"J:\30g\l_pqesod2" size 231000;
create dbslice l_pqesod19 from cogroup ifmx chunk
"K:\30g\l_pqesod2" size 231000;
create dbslice l_pqesod20 from cogroup ifmx chunk
"L:\30g\l_pqesod2" size 231000;
create dbslice l_pqesod21 from cogroup ifmx chunk
"M:\30g\l_pqesod2" size 231000;
create dbslice l_pqesod22 from cogroup ifmx chunk
"n:\30g\l_pqesod2" size 231000;
create dbslice l_pqesod23 from cogroup ifmx chunk
"o:\30g\l_pqesod2" size 231000;
create dbslice l_pqesod24 from cogroup ifmx chunk
"p:\30g\l_pqesod2" size 231000;
create dbslice l_pqesod25 from cogroup ifmx chunk
"q:\30g\l_pqesod2" size 231000;
create dbslice l_pqesod26 from cogroup ifmx chunk
"r:\30g\l_pqesod2" size 231000;
create dbslice l_pqesod27 from cogroup ifmx chunk
"s:\30g\l_pqesod2" size 231000;
create dbslice l_pqesod28 from cogroup ifmx chunk
"t:\30g\l_pqesod2" size 231000;
```

B-17: create_tables.sql

```
drop table lineitem;
create operational table lineitem
(
l_orderkey integer,
l_partkey integer ,
l_suppkey integer ,
l_linenumber integer ,
l_quantity decimal(12,2) NOT NULL,
l_extendedprice decimal(12,2) NOT NULL,
l_discount decimal(12,2) NOT NULL,
l_tax decimal(12,2) ,
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)
) fragment by hybrid(l_orderkey) expression
l_shipdate < '1992-02-01' in l_month1,
l_shipdate >= '1992-02-01' and l_shipdate < '1992-03-01' in
l_month2,
l_shipdate >= '1992-03-01' and l_shipdate < '1992-04-01' in
l_month3,
l_shipdate >= '1992-04-01' and l_shipdate < '1992-05-01' in
l_month4,
l_shipdate >= '1992-05-01' and l_shipdate < '1992-06-01' in
l_month5,
l_shipdate >= '1992-06-01' and l_shipdate < '1992-07-01' in
l_month6,
l_shipdate >= '1992-07-01' and l_shipdate < '1992-08-01' in
l_month7,
l_shipdate >= '1992-08-01' and l_shipdate < '1992-09-01' in
l_month8,
```

```
l_shipdate >= '1992-09-01' and l_shipdate < '1992-10-01' in
l_month9,
l_shipdate >= '1992-10-01' and l_shipdate < '1992-11-01' in
l_month10,
l_shipdate >= '1992-11-01' and l_shipdate < '1992-12-01' in
l_month11,
l_shipdate >= '1992-12-01' and l_shipdate < '1993-01-01' in
l_month12,
l_shipdate >= '1993-01-01' and l_shipdate < '1993-02-01' in
l_month13,
l_shipdate >= '1993-02-01' and l_shipdate < '1993-03-01' in
l_month14,
l_shipdate >= '1993-03-01' and l_shipdate < '1993-04-01' in
l_month15,
l_shipdate >= '1993-04-01' and l_shipdate < '1993-05-01' in
l_month16,
l_shipdate >= '1993-05-01' and l_shipdate < '1993-06-01' in
l_month17,
l_shipdate >= '1993-06-01' and l_shipdate < '1993-07-01' in
l_month18,
l_shipdate >= '1993-07-01' and l_shipdate < '1993-08-01' in
l_month19,
l_shipdate >= '1993-08-01' and l_shipdate < '1993-09-01' in
l_month20,
l_shipdate >= '1993-09-01' and l_shipdate < '1993-10-01' in
l_month21,
l_shipdate >= '1993-10-01' and l_shipdate < '1993-11-01' in
l_month22,
l_shipdate >= '1993-11-01' and l_shipdate < '1993-12-01' in
l_month23,
l_shipdate >= '1993-12-01' and l_shipdate < '1994-01-01' in
l_month24,
l_shipdate >= '1994-01-01' and l_shipdate < '1994-02-01' in
l_month25,
l_shipdate >= '1994-02-01' and l_shipdate < '1994-03-01' in
l_month26,
l_shipdate >= '1994-03-01' and l_shipdate < '1994-04-01' in
l_month27,
l_shipdate >= '1994-04-01' and l_shipdate < '1994-05-01' in
l_month28,
l_shipdate >= '1994-05-01' and l_shipdate < '1994-06-01' in
l_month29,
l_shipdate >= '1994-06-01' and l_shipdate < '1994-07-01' in
l_month30,
l_shipdate >= '1994-07-01' and l_shipdate < '1994-08-01' in
l_month31,
l_shipdate >= '1994-08-01' and l_shipdate < '1994-09-01' in
l_month32,
l_shipdate >= '1994-09-01' and l_shipdate < '1994-10-01' in
l_month33,
l_shipdate >= '1994-10-01' and l_shipdate < '1994-11-01' in
l_month34,
l_shipdate >= '1994-11-01' and l_shipdate < '1994-12-01' in
l_month35,
l_shipdate >= '1994-12-01' and l_shipdate < '1995-01-01' in
l_month36,
l_shipdate >= '1995-01-01' and l_shipdate < '1995-02-01' in
l_month37,
l_shipdate >= '1995-02-01' and l_shipdate < '1995-03-01' in
l_month38,
l_shipdate >= '1995-03-01' and l_shipdate < '1995-04-01' in
l_month39,
l_shipdate >= '1995-04-01' and l_shipdate < '1995-05-01' in
l_month40,
l_shipdate >= '1995-05-01' and l_shipdate < '1995-06-01' in
l_month41,
l_shipdate >= '1995-06-01' and l_shipdate < '1995-07-01' in
l_month42,
l_shipdate >= '1995-07-01' and l_shipdate < '1995-08-01' in
l_month43,
l_shipdate >= '1995-08-01' and l_shipdate < '1995-09-01' in
l_month44,
l_shipdate >= '1995-09-01' and l_shipdate < '1995-10-01' in
l_month45,
l_shipdate >= '1995-10-01' and l_shipdate < '1995-11-01' in
l_month46,
l_shipdate >= '1995-11-01' and l_shipdate < '1995-12-01' in
l_month47,
l_shipdate >= '1995-12-01' and l_shipdate < '1996-01-01' in
l_month48,
l_shipdate >= '1996-01-01' and l_shipdate < '1996-02-01' in
l_month49,
l_shipdate >= '1996-02-01' and l_shipdate < '1996-03-01' in
l_month50,
l_shipdate >= '1996-03-01' and l_shipdate < '1996-04-01' in
l_month51,
l_shipdate >= '1996-04-01' and l_shipdate < '1996-05-01' in
l_month52,
l_shipdate >= '1996-05-01' and l_shipdate < '1996-06-01' in
l_month53,
l_shipdate >= '1996-06-01' and l_shipdate < '1996-07-01' in
l_month54,
l_shipdate >= '1996-07-01' and l_shipdate < '1996-08-01' in
l_month55,
l_shipdate >= '1996-08-01' and l_shipdate < '1996-09-01' in
l_month56,
l_shipdate >= '1996-09-01' and l_shipdate < '1996-10-01' in
l_month57,
l_shipdate >= '1996-10-01' and l_shipdate < '1996-11-01' in
l_month58,
l_shipdate >= '1996-11-01' and l_shipdate < '1996-12-01' in
l_month59,
l_shipdate >= '1996-12-01' and l_shipdate < '1997-01-01' in
l_month60,
l_shipdate >= '1997-01-01' and l_shipdate < '1997-02-01' in
l_month61,
l_shipdate >= '1997-02-01' and l_shipdate < '1997-03-01' in
l_month62,
l_shipdate >= '1997-03-01' and l_shipdate < '1997-04-01' in
l_month63,
l_shipdate >= '1997-04-01' and l_shipdate < '1997-05-01' in
l_month64,
l_shipdate >= '1997-05-01' and l_shipdate < '1997-06-01' in
l_month65,
l_shipdate >= '1997-06-01' and l_shipdate < '1997-07-01' in
l_month66,
l_shipdate >= '1997-07-01' and l_shipdate < '1997-08-01' in
l_month67,
l_shipdate >= '1997-08-01' and l_shipdate < '1997-09-01' in
l_month68,
l_shipdate >= '1997-09-01' and l_shipdate < '1997-10-01' in
l_month69,
l_shipdate >= '1997-10-01' and l_shipdate < '1997-11-01' in
l_month70,
l_shipdate >= '1997-11-01' and l_shipdate < '1997-12-01' in
l_month71,
l_shipdate >= '1997-12-01' and l_shipdate < '1998-01-01' in
l_month72,
l_shipdate >= '1998-01-01' and l_shipdate < '1998-02-01' in
l_month73,
l_shipdate >= '1998-02-01' and l_shipdate < '1998-03-01' in
l_month74,
l_shipdate >= '1998-03-01' and l_shipdate < '1998-04-01' in
l_month75,
l_shipdate >= '1998-04-01' and l_shipdate < '1998-05-01' in
l_month76,
l_shipdate >= '1998-05-01' and l_shipdate < '1998-06-01' in
l_month77,
l_shipdate >= '1998-06-01' and l_shipdate < '1998-07-01' in
l_month78,
l_shipdate >= '1998-07-01' and l_shipdate < '1998-08-01' in
l_month79,
```

```
l_shipdate >= '1998-08-01' and l_shipdate < '1998-09-01' in
l_month80,
l_shipdate >= '1998-09-01' and l_shipdate < '1998-10-01' in
l_month81,
l_shipdate >= '1998-10-01' and l_shipdate < '1998-11-01' in
l_month82,
l_shipdate >= '1998-11-01' and l_shipdate < '1998-12-01' in
l_month83,
l_shipdate >= '1998-12-01'  in l_month84
extent size  330000 next size 2000
lock mode table;

drop table order;
create operational table order
(
o_orderkey integer,
o_custkey integer ,
o_orderstatus char(1) ,
o_totalprice decimal(12,2) ,
o_orderdate date,
o_orderpriority char(15) ,
o_clerk char(15),
o_shippriority integer ,
o_comment varchar(79)
) fragment by hybrid(o_orderkey) expression
o_orderdate < '1992-02-01' in o_month1,
o_orderdate >= '1992-02-01' and o_orderdate < '1992-03-01'
in o_month2,
o_orderdate >= '1992-03-01' and o_orderdate < '1992-04-01'
in o_month3,
o_orderdate >= '1992-04-01' and o_orderdate < '1992-05-01'
in o_month4,
o_orderdate >= '1992-05-01' and o_orderdate < '1992-06-01'
in o_month5,
o_orderdate >= '1992-06-01' and o_orderdate < '1992-07-01'
in o_month6,
o_orderdate >= '1992-07-01' and o_orderdate < '1992-08-01'
in o_month7,
o_orderdate >= '1992-08-01' and o_orderdate < '1992-09-01'
in o_month8,
o_orderdate >= '1992-09-01' and o_orderdate < '1992-10-01'
in o_month9,
o_orderdate >= '1992-10-01' and o_orderdate < '1992-11-01'
in o_month10,
o_orderdate >= '1992-11-01' and o_orderdate < '1992-12-01'
in o_month11,
o_orderdate >= '1992-12-01' and o_orderdate < '1993-01-01'
in o_month12,
o_orderdate >= '1993-01-01' and o_orderdate < '1993-02-01'
in o_month13,
o_orderdate >= '1993-02-01' and o_orderdate < '1993-03-01'
in o_month14,
o_orderdate >= '1993-03-01' and o_orderdate < '1993-04-01'
in o_month15,
o_orderdate >= '1993-04-01' and o_orderdate < '1993-05-01'
in o_month16,
o_orderdate >= '1993-05-01' and o_orderdate < '1993-06-01'
in o_month17,
o_orderdate >= '1993-06-01' and o_orderdate < '1993-07-01'
in o_month18,
o_orderdate >= '1993-07-01' and o_orderdate < '1993-08-01'
in o_month19,
o_orderdate >= '1993-08-01' and o_orderdate < '1993-09-01'
in o_month20,
o_orderdate >= '1993-09-01' and o_orderdate < '1993-10-01'
in o_month21,
o_orderdate >= '1993-10-01' and o_orderdate < '1993-11-01'
in o_month22,
o_orderdate >= '1993-11-01' and o_orderdate < '1993-12-01'
in o_month23,
o_orderdate >= '1993-12-01' and o_orderdate < '1994-01-01'
in o_month24,
o_orderdate >= '1994-01-01' and o_orderdate < '1994-02-01'
in o_month25,
o_orderdate >= '1994-02-01' and o_orderdate < '1994-03-01'
in o_month26,
o_orderdate >= '1994-03-01' and o_orderdate < '1994-04-01'
in o_month27,
o_orderdate >= '1994-04-01' and o_orderdate < '1994-05-01'
in o_month28,
o_orderdate >= '1994-05-01' and o_orderdate < '1994-06-01'
in o_month29,
o_orderdate >= '1994-06-01' and o_orderdate < '1994-07-01'
in o_month30,
o_orderdate >= '1994-07-01' and o_orderdate < '1994-08-01'
in o_month31,
o_orderdate >= '1994-08-01' and o_orderdate < '1994-09-01'
in o_month32,
o_orderdate >= '1994-09-01' and o_orderdate < '1994-10-01'
in o_month33,
o_orderdate >= '1994-10-01' and o_orderdate < '1994-11-01'
in o_month34,
o_orderdate >= '1994-11-01' and o_orderdate < '1994-12-01'
in o_month35,
o_orderdate >= '1994-12-01' and o_orderdate < '1995-01-01'
in o_month36,
o_orderdate >= '1995-01-01' and o_orderdate < '1995-02-01'
in o_month37,
o_orderdate >= '1995-02-01' and o_orderdate < '1995-03-01'
in o_month38,
o_orderdate >= '1995-03-01' and o_orderdate < '1995-04-01'
in o_month39,
o_orderdate >= '1995-04-01' and o_orderdate < '1995-05-01'
in o_month40,
o_orderdate >= '1995-05-01' and o_orderdate < '1995-06-01'
in o_month41,
o_orderdate >= '1995-06-01' and o_orderdate < '1995-07-01'
in o_month42,
o_orderdate >= '1995-07-01' and o_orderdate < '1995-08-01'
in o_month43,
o_orderdate >= '1995-08-01' and o_orderdate < '1995-09-01'
in o_month44,
o_orderdate >= '1995-09-01' and o_orderdate < '1995-10-01'
in o_month45,
o_orderdate >= '1995-10-01' and o_orderdate < '1995-11-01'
in o_month46,
o_orderdate >= '1995-11-01' and o_orderdate < '1995-12-01'
in o_month47,
o_orderdate >= '1995-12-01' and o_orderdate < '1996-01-01'
in o_month48,
o_orderdate >= '1996-01-01' and o_orderdate < '1996-02-01'
in o_month49,
o_orderdate >= '1996-02-01' and o_orderdate < '1996-03-01'
in o_month50,
o_orderdate >= '1996-03-01' and o_orderdate < '1996-04-01'
in o_month51,
o_orderdate >= '1996-04-01' and o_orderdate < '1996-05-01'
in o_month52,
o_orderdate >= '1996-05-01' and o_orderdate < '1996-06-01'
in o_month53,
o_orderdate >= '1996-06-01' and o_orderdate < '1996-07-01'
in o_month54,
o_orderdate >= '1996-07-01' and o_orderdate < '1996-08-01'
in o_month55,
o_orderdate >= '1996-08-01' and o_orderdate < '1996-09-01'
in o_month56,
o_orderdate >= '1996-09-01' and o_orderdate < '1996-10-01'
in o_month57,
o_orderdate >= '1996-10-01' and o_orderdate < '1996-11-01'
in o_month58,
o_orderdate >= '1996-11-01' and o_orderdate < '1996-12-01'
in o_month59,
```

```
o_orderdate >= '1996-12-01' and o_orderdate < '1997-01-01'
in o_month60,
o_orderdate >= '1997-01-01' and o_orderdate < '1997-02-01'
in o_month61,
o_orderdate >= '1997-02-01' and o_orderdate < '1997-03-01'
in o_month62,
o_orderdate >= '1997-03-01' and o_orderdate < '1997-04-01'
in o_month63,
o_orderdate >= '1997-04-01' and o_orderdate < '1997-05-01'
in o_month64,
o_orderdate >= '1997-05-01' and o_orderdate < '1997-06-01'
in o_month65,
o_orderdate >= '1997-06-01' and o_orderdate < '1997-07-01'
in o_month66,
o_orderdate >= '1997-07-01' and o_orderdate < '1997-08-01'
in o_month67,
o_orderdate >= '1997-08-01' and o_orderdate < '1997-09-01'
in o_month68,
o_orderdate >= '1997-09-01' and o_orderdate < '1997-10-01'
in o_month69,
o_orderdate >= '1997-10-01' and o_orderdate < '1997-11-01'
in o_month70,
o_orderdate >= '1997-11-01' and o_orderdate < '1997-12-01'
in o_month71,
o_orderdate >= '1997-12-01' and o_orderdate < '1998-01-01'
in o_month72,
o_orderdate >= '1998-01-01' and o_orderdate < '1998-02-01'
in o_month73,
o_orderdate >= '1998-02-01' and o_orderdate < '1998-03-01'
in o_month74,
o_orderdate >= '1998-03-01' and o_orderdate < '1998-04-01'
in o_month75,
o_orderdate >= '1998-04-01' and o_orderdate < '1998-05-01'
in o_month76,
o_orderdate >= '1998-05-01' and o_orderdate < '1998-06-01'
in o_month77,
o_orderdate >= '1998-06-01' and o_orderdate < '1998-07-01'
in o_month78,
o_orderdate >= '1998-07-01' and o_orderdate < '1998-08-01'
in o_month79,
o_orderdate >= '1998-08-01' and o_orderdate < '1998-09-01'
in o_month80,
o_orderdate >= '1998-09-01' and o_orderdate < '1998-10-01'
in o_month81,
o_orderdate >= '1998-10-01' and o_orderdate < '1998-11-01'
in o_month82,
o_orderdate >= '1998-11-01' and o_orderdate < '1998-12-01'
in o_month83,
o_orderdate >= '1998-12-01' in o_month84
extent size  78000 next size 2000
lock mode table;

drop table customer;
create operational table customer
(
c_custkey integer,
c_name varchar(25),
c_address varchar(40) ,
c_nationkey integer ,
c_phone char(15) ,
c_acctbal decimal(12,2) ,
c_mktsegment char(10) ,
c_comment varchar(117)
) fragment by hash(c_custkey) in cust
extent size  36000 next size 1000
lock mode table;

drop table part;
create operational table part
(
p_partkey integer,
p_name varchar(55),
p_mfgr char(25),
p_brand char(10) ,
p_type varchar(25) ,
p_size integer ,
p_container char(10) ,
p_retailprice decimal(12,2) ,
p_comment varchar(23)
) fragment by hash(p_partkey) in part
extent size  36000 next size 1000
lock mode table;

drop table supplier;
create operational table supplier
(
s_suppkey integer,
s_name char(25),
s_address varchar(40),
s_nationkey integer,
s_phone char(15),
s_acctbal decimal(12,2),
s_comment varchar(101)
) fragment by hash(s_suppkey) in supp
extent size  8000 next size 1000
lock mode table;

drop table partsupp;
create operational table partsupp
(
ps_partkey integer,
ps_suppkey integer ,
ps_availqty integer ,
ps_supplycost decimal(12,2) ,
ps_comment varchar(199)
) fragment by hash(ps_partkey) in ps_supp1
extent size 180000 next size 2000
lock mode table;

drop table nation;
create operational table nation
(
n_nationkey integer,
n_name char(25),
n_regionkey integer,
n_comment varchar(152)
) in supp.1
extent size 16 next size 16
lock mode table;

drop table region;
create operational table region
(
r_regionkey integer,
r_name char(25),
r_comment varchar(152)
) in supp.1
extent size 16 next size 16
lock mode table;
grant select on customer to "public";
grant update on customer to "public";
grant insert on customer to "public";
grant delete on customer to "public";
grant index on customer to "public";
grant select on order to "public";
grant update on order to "public";
grant insert on order to "public";
grant delete on order to "public";
grant index on order to "public";
grant select on lineitem to "public";
grant update on lineitem to "public";
grant insert on lineitem to "public";
```

```
grant delete on lineitem to "public";
grant index on lineitem to "public";
grant select on part to "public";
grant update on part to "public";
grant insert on part to "public";
grant delete on part to "public";
grant index on part to "public";
grant select on supplier to "public";
grant update on supplier to "public";
grant insert on supplier to "public";
grant delete on supplier to "public";
grant index on supplier to "public";
grant select on partsupp to "public";
grant update on partsupp to "public";
grant insert on partsupp to "public";
grant delete on partsupp to "public";
grant index on partsupp to "public";
grant select on nation to "public";
grant update on nation to "public";
grant insert on nation to "public";
grant delete on nation to "public";
grant index on nation to "public";
grant select on region to "public";
grant update on region to "public";
grant insert on region to "public";
grant delete on region to "public";
grant index on region to "public";
```

B-18: load_tables.sql

```
alter table lineitem type (raw);
create external table lineitem_ext
sameas lineitem
using (
format "delimited",
datafiles (
"disk:1:\\inform3\G$\30G\lineitem.tbl.1",
"disk:1:\\inform3\G$\30G\lineitem.tbl.2",
"disk:1:\\inform3\G$\30G\lineitem.tbl.3",
"disk:1:\\inform3\G$\30G\lineitem.tbl.4",
"disk:1:\\inform3\G$\30G\lineitem.tbl.5",
"disk:1:\\inform3\G$\30G\lineitem.tbl.6",
"disk:1:\\inform3\G$\30G\lineitem.tbl.7",
"disk:1:\\inform3\G$\30G\lineitem.tbl.8",
"disk:1:\\inform3\G$\30G\lineitem.tbl.9",
"disk:1:\\inform3\G$\30G\lineitem.tbl.10",
"disk:1:\\inform3\H$\30G\lineitem.tbl.11",
"disk:1:\\inform3\H$\30G\lineitem.tbl.12",
"disk:1:\\inform3\H$\30G\lineitem.tbl.13",
"disk:1:\\inform3\H$\30G\lineitem.tbl.14",
"disk:1:\\inform3\H$\30G\lineitem.tbl.15",
"disk:1:\\inform3\H$\30G\lineitem.tbl.16",
"disk:1:\\inform3\H$\30G\lineitem.tbl.17",
"disk:1:\\inform3\H$\30G\lineitem.tbl.18",
"disk:1:\\inform3\H$\30G\lineitem.tbl.19",
"disk:1:\\inform3\H$\30G\lineitem.tbl.20",
"disk:1:\\inform3\I$\30G\lineitem.tbl.21",
"disk:1:\\inform3\I$\30G\lineitem.tbl.22",
"disk:1:\\inform3\I$\30G\lineitem.tbl.23",
"disk:1:\\inform3\I$\30G\lineitem.tbl.24",
"disk:1:\\inform3\I$\30G\lineitem.tbl.25",
"disk:1:\\inform3\I$\30G\lineitem.tbl.26",
"disk:1:\\inform3\I$\30G\lineitem.tbl.27",
"disk:1:\\inform3\I$\30G\lineitem.tbl.28",
"disk:1:\\inform3\I$\30G\lineitem.tbl.29",
"disk:1:\\inform3\I$\30G\lineitem.tbl.30",
"disk:1:\\inform3\J$\30G\lineitem.tbl.31",
"disk:1:\\inform3\J$\30G\lineitem.tbl.32",
"disk:1:\\inform3\J$\30G\lineitem.tbl.33",
"disk:1:\\inform3\J$\30G\lineitem.tbl.34",
"disk:1:\\inform3\J$\30G\lineitem.tbl.35",
"disk:1:\\inform3\J$\30G\lineitem.tbl.36",
"disk:1:\\inform3\J$\30G\lineitem.tbl.37",
"disk:1:\\inform3\J$\30G\lineitem.tbl.38",
"disk:1:\\inform3\J$\30G\lineitem.tbl.39",
"disk:1:\\inform3\J$\30G\lineitem.tbl.40",
"disk:1:\\inform3\K$\30G\lineitem.tbl.41",
"disk:1:\\inform3\K$\30G\lineitem.tbl.42",
"disk:1:\\inform3\K$\30G\lineitem.tbl.43",
"disk:1:\\inform3\K$\30G\lineitem.tbl.44",
"disk:1:\\inform3\K$\30G\lineitem.tbl.45",
"disk:1:\\inform3\K$\30G\lineitem.tbl.46",
"disk:1:\\inform3\K$\30G\lineitem.tbl.47",
"disk:1:\\inform3\K$\30G\lineitem.tbl.48",
"disk:1:\\inform3\K$\30G\lineitem.tbl.49",
"disk:1:\\inform3\K$\30G\lineitem.tbl.50",
"disk:1:\\inform3\L$\30G\lineitem.tbl.51",
"disk:1:\\inform3\L$\30G\lineitem.tbl.52",
"disk:1:\\inform3\L$\30G\lineitem.tbl.53",
"disk:1:\\inform3\L$\30G\lineitem.tbl.54",
"disk:1:\\inform3\L$\30G\lineitem.tbl.55",
"disk:1:\\inform3\L$\30G\lineitem.tbl.56",
"disk:1:\\inform3\L$\30G\lineitem.tbl.57",
"disk:1:\\inform3\L$\30G\lineitem.tbl.58",
"disk:1:\\inform3\L$\30G\lineitem.tbl.59",
"disk:1:\\inform3\L$\30G\lineitem.tbl.60",
"disk:1:\\inform3\M$\30G\lineitem.tbl.61",
"disk:1:\\inform3\M$\30G\lineitem.tbl.62",
"disk:1:\\inform3\M$\30G\lineitem.tbl.63",
"disk:1:\\inform3\M$\30G\lineitem.tbl.64",
"disk:1:\\inform3\M$\30G\lineitem.tbl.65",
"disk:1:\\inform3\M$\30G\lineitem.tbl.66",
"disk:1:\\inform3\M$\30G\lineitem.tbl.67",
"disk:1:\\inform3\M$\30G\lineitem.tbl.68",
"disk:1:\\inform3\M$\30G\lineitem.tbl.69",
"disk:1:\\inform3\M$\30G\lineitem.tbl.70",
"disk:1:\\inform3\N$\30G\lineitem.tbl.71",
"disk:1:\\inform3\N$\30G\lineitem.tbl.72",
"disk:1:\\inform3\N$\30G\lineitem.tbl.73",
"disk:1:\\inform3\N$\30G\lineitem.tbl.74",
"disk:1:\\inform3\N$\30G\lineitem.tbl.75",
"disk:1:\\inform3\N$\30G\lineitem.tbl.76",
"disk:1:\\inform3\N$\30G\lineitem.tbl.77",
"disk:1:\\inform3\N$\30G\lineitem.tbl.78",
"disk:1:\\inform3\N$\30G\lineitem.tbl.79",
"disk:1:\\inform3\N$\30G\lineitem.tbl.80"
),
rejectfile "\\inform3\N$\30G\lineitem%c.rej",
express
);
insert into lineitem select * from lineitem_ext;
drop table lineitem_ext;

alter table order type (raw);
create external table order_ext
sameas order
using (
format "delimited",
datafiles (
"disk:1:\\inform3\G$\30G\order.tbl.1",
"disk:1:\\inform3\G$\30G\order.tbl.2",
"disk:1:\\inform3\G$\30G\order.tbl.3",
"disk:1:\\inform3\G$\30G\order.tbl.4",
"disk:1:\\inform3\G$\30G\order.tbl.5",
"disk:1:\\inform3\G$\30G\order.tbl.6",
"disk:1:\\inform3\G$\30G\order.tbl.7",
"disk:1:\\inform3\G$\30G\order.tbl.8",
"disk:1:\\inform3\G$\30G\order.tbl.9",
"disk:1:\\inform3\G$\30G\order.tbl.10",
"disk:1:\\inform3\H$\30G\order.tbl.11",
"disk:1:\\inform3\H$\30G\order.tbl.12",
```

```
"disk:1:\\inform3\H$\30G\order.tbl.13",
"disk:1:\\inform3\H$\30G\order.tbl.14",
"disk:1:\\inform3\H$\30G\order.tbl.15",
"disk:1:\\inform3\H$\30G\order.tbl.16",
"disk:1:\\inform3\H$\30G\order.tbl.17",
"disk:1:\\inform3\H$\30G\order.tbl.18",
"disk:1:\\inform3\H$\30G\order.tbl.19",
"disk:1:\\inform3\H$\30G\order.tbl.20",
"disk:1:\\inform3\I$\30G\order.tbl.21",
"disk:1:\\inform3\I$\30G\order.tbl.22",
"disk:1:\\inform3\I$\30G\order.tbl.23",
"disk:1:\\inform3\I$\30G\order.tbl.24",
"disk:1:\\inform3\I$\30G\order.tbl.25",
"disk:1:\\inform3\I$\30G\order.tbl.26",
"disk:1:\\inform3\I$\30G\order.tbl.27",
"disk:1:\\inform3\I$\30G\order.tbl.28",
"disk:1:\\inform3\I$\30G\order.tbl.29",
"disk:1:\\inform3\I$\30G\order.tbl.30",
"disk:1:\\inform3\J$\30G\order.tbl.31",
"disk:1:\\inform3\J$\30G\order.tbl.32",
"disk:1:\\inform3\J$\30G\order.tbl.33",
"disk:1:\\inform3\J$\30G\order.tbl.34",
"disk:1:\\inform3\J$\30G\order.tbl.35",
"disk:1:\\inform3\J$\30G\order.tbl.36",
"disk:1:\\inform3\J$\30G\order.tbl.37",
"disk:1:\\inform3\J$\30G\order.tbl.38",
"disk:1:\\inform3\J$\30G\order.tbl.39",
"disk:1:\\inform3\J$\30G\order.tbl.40",
"disk:1:\\inform3\K$\30G\order.tbl.41",
"disk:1:\\inform3\K$\30G\order.tbl.42",
"disk:1:\\inform3\K$\30G\order.tbl.43",
"disk:1:\\inform3\K$\30G\order.tbl.44",
"disk:1:\\inform3\K$\30G\order.tbl.45",
"disk:1:\\inform3\K$\30G\order.tbl.46",
"disk:1:\\inform3\K$\30G\order.tbl.47",
"disk:1:\\inform3\K$\30G\order.tbl.48",
"disk:1:\\inform3\K$\30G\order.tbl.49",
"disk:1:\\inform3\K$\30G\order.tbl.50",
"disk:1:\\inform3\L$\30G\order.tbl.51",
"disk:1:\\inform3\L$\30G\order.tbl.52",
"disk:1:\\inform3\L$\30G\order.tbl.53",
"disk:1:\\inform3\L$\30G\order.tbl.54",
"disk:1:\\inform3\L$\30G\order.tbl.55",
"disk:1:\\inform3\L$\30G\order.tbl.56",
"disk:1:\\inform3\L$\30G\order.tbl.57",
"disk:1:\\inform3\L$\30G\order.tbl.58",
"disk:1:\\inform3\L$\30G\order.tbl.59",
"disk:1:\\inform3\L$\30G\order.tbl.60",
"disk:1:\\inform3\M$\30G\order.tbl.61",
"disk:1:\\inform3\M$\30G\order.tbl.62",
"disk:1:\\inform3\M$\30G\order.tbl.63",
"disk:1:\\inform3\M$\30G\order.tbl.64",
"disk:1:\\inform3\M$\30G\order.tbl.65",
"disk:1:\\inform3\M$\30G\order.tbl.66",
"disk:1:\\inform3\M$\30G\order.tbl.67",
"disk:1:\\inform3\M$\30G\order.tbl.68",
"disk:1:\\inform3\M$\30G\order.tbl.69",
"disk:1:\\inform3\M$\30G\order.tbl.70",
"disk:1:\\inform3\N$\30G\order.tbl.71",
"disk:1:\\inform3\N$\30G\order.tbl.72",
"disk:1:\\inform3\N$\30G\order.tbl.73",
"disk:1:\\inform3\N$\30G\order.tbl.74",
"disk:1:\\inform3\N$\30G\order.tbl.75",
"disk:1:\\inform3\N$\30G\order.tbl.76",
"disk:1:\\inform3\N$\30G\order.tbl.77",
"disk:1:\\inform3\N$\30G\order.tbl.78",
"disk:1:\\inform3\N$\30G\order.tbl.79",
"disk:1:\\inform3\N$\30G\order.tbl.80"
),
rejectfile "\\inform3\N$\30G\order%c.rej",
express
);
insert into order select * from order_ext;
drop table order_ext;

alter table customer type (raw);
create external table customer_ext
sameas customer
using (
format "delimited",
datafiles (
"disk:1:\\inform3\G$\30G\customer.tbl.1",
"disk:1:\\inform3\G$\30G\customer.tbl.2",
"disk:1:\\inform3\G$\30G\customer.tbl.3",
"disk:1:\\inform3\G$\30G\customer.tbl.4",
"disk:1:\\inform3\G$\30G\customer.tbl.5",
"disk:1:\\inform3\G$\30G\customer.tbl.6",
"disk:1:\\inform3\G$\30G\customer.tbl.7",
"disk:1:\\inform3\G$\30G\customer.tbl.8",
"disk:1:\\inform3\G$\30G\customer.tbl.9",
"disk:1:\\inform3\G$\30G\customer.tbl.10",
"disk:1:\\inform3\H$\30G\customer.tbl.11",
"disk:1:\\inform3\H$\30G\customer.tbl.12",
"disk:1:\\inform3\H$\30G\customer.tbl.13",
"disk:1:\\inform3\H$\30G\customer.tbl.14",
"disk:1:\\inform3\H$\30G\customer.tbl.15",
"disk:1:\\inform3\H$\30G\customer.tbl.16",
"disk:1:\\inform3\H$\30G\customer.tbl.17",
"disk:1:\\inform3\H$\30G\customer.tbl.18",
"disk:1:\\inform3\H$\30G\customer.tbl.19",
"disk:1:\\inform3\H$\30G\customer.tbl.20",
"disk:1:\\inform3\I$\30G\customer.tbl.21",
"disk:1:\\inform3\I$\30G\customer.tbl.22",
"disk:1:\\inform3\I$\30G\customer.tbl.23",
"disk:1:\\inform3\I$\30G\customer.tbl.24",
"disk:1:\\inform3\I$\30G\customer.tbl.25",
"disk:1:\\inform3\I$\30G\customer.tbl.26",
"disk:1:\\inform3\I$\30G\customer.tbl.27",
"disk:1:\\inform3\I$\30G\customer.tbl.28",
"disk:1:\\inform3\I$\30G\customer.tbl.29",
"disk:1:\\inform3\I$\30G\customer.tbl.30",
"disk:1:\\inform3\J$\30G\customer.tbl.31",
"disk:1:\\inform3\J$\30G\customer.tbl.32",
"disk:1:\\inform3\J$\30G\customer.tbl.33",
"disk:1:\\inform3\J$\30G\customer.tbl.34",
"disk:1:\\inform3\J$\30G\customer.tbl.35",
"disk:1:\\inform3\J$\30G\customer.tbl.36",
"disk:1:\\inform3\J$\30G\customer.tbl.37",
"disk:1:\\inform3\J$\30G\customer.tbl.38",
"disk:1:\\inform3\J$\30G\customer.tbl.39",
"disk:1:\\inform3\J$\30G\customer.tbl.40",
"disk:1:\\inform3\K$\30G\customer.tbl.41",
"disk:1:\\inform3\K$\30G\customer.tbl.42",
"disk:1:\\inform3\K$\30G\customer.tbl.43",
"disk:1:\\inform3\K$\30G\customer.tbl.44",
"disk:1:\\inform3\K$\30G\customer.tbl.45",
"disk:1:\\inform3\K$\30G\customer.tbl.46",
"disk:1:\\inform3\K$\30G\customer.tbl.47",
"disk:1:\\inform3\K$\30G\customer.tbl.48",
"disk:1:\\inform3\K$\30G\customer.tbl.49",
"disk:1:\\inform3\K$\30G\customer.tbl.50",
"disk:1:\\inform3\L$\30G\customer.tbl.51",
"disk:1:\\inform3\L$\30G\customer.tbl.52",
"disk:1:\\inform3\L$\30G\customer.tbl.53",
"disk:1:\\inform3\L$\30G\customer.tbl.54",
"disk:1:\\inform3\L$\30G\customer.tbl.55",
"disk:1:\\inform3\L$\30G\customer.tbl.56",
"disk:1:\\inform3\L$\30G\customer.tbl.57",
"disk:1:\\inform3\L$\30G\customer.tbl.58",
"disk:1:\\inform3\L$\30G\customer.tbl.59",
"disk:1:\\inform3\L$\30G\customer.tbl.60",
"disk:1:\\inform3\M$\30G\customer.tbl.61",
```

```
"disk:1:\\inform3\M$\30G\customer.tbl.62",
"disk:1:\\inform3\M$\30G\customer.tbl.63",
"disk:1:\\inform3\M$\30G\customer.tbl.64",
"disk:1:\\inform3\M$\30G\customer.tbl.65",
"disk:1:\\inform3\M$\30G\customer.tbl.66",
"disk:1:\\inform3\M$\30G\customer.tbl.67",
"disk:1:\\inform3\M$\30G\customer.tbl.68",
"disk:1:\\inform3\M$\30G\customer.tbl.69",
"disk:1:\\inform3\M$\30G\customer.tbl.70",
"disk:1:\\inform3\N$\30G\customer.tbl.71",
"disk:1:\\inform3\N$\30G\customer.tbl.72",
"disk:1:\\inform3\N$\30G\customer.tbl.73",
"disk:1:\\inform3\N$\30G\customer.tbl.74",
"disk:1:\\inform3\N$\30G\customer.tbl.75",
"disk:1:\\inform3\N$\30G\customer.tbl.76",
"disk:1:\\inform3\N$\30G\customer.tbl.77",
"disk:1:\\inform3\N$\30G\customer.tbl.78",
"disk:1:\\inform3\N$\30G\customer.tbl.79",
"disk:1:\\inform3\N$\30G\customer.tbl.80"
),
rejectfile "\\inform3\N$\30G\customer%c.rej",
express
);
insert into customer select * from customer_ext;
drop table customer_ext;

alter table region type (raw);
create external table region_ext
sameas region
using (
format "delimited",
datafiles ("disk:1:\\inform3\N$\30G\region.tbl"),
rejectfile "\\inform3\N$\30G\region%c.rej",
express
);
insert into region select * from region_ext;
drop table region_ext;

alter table nation type (raw);
create external table nation_ext
sameas nation
using (
format "delimited",
datafiles ("disk:1:\\inform3\N$\30G\nation.tbl"),
rejectfile "\\inform3\N$\30G\nation%c.rej",
express
);
insert into nation select * from nation_ext;
drop table nation_ext;

alter table supplier type (raw);
create external table supp_ext
sameas supplier
using (
format "delimited",
datafiles (
"disk:1:\\inform3\G$\30G\supplier.tbl.1",
"disk:1:\\inform3\G$\30G\supplier.tbl.2",
"disk:1:\\inform3\G$\30G\supplier.tbl.3",
"disk:1:\\inform3\G$\30G\supplier.tbl.4",
"disk:1:\\inform3\G$\30G\supplier.tbl.5",
"disk:1:\\inform3\G$\30G\supplier.tbl.6",
"disk:1:\\inform3\G$\30G\supplier.tbl.7",
"disk:1:\\inform3\G$\30G\supplier.tbl.8",
"disk:1:\\inform3\G$\30G\supplier.tbl.9",
"disk:1:\\inform3\G$\30G\supplier.tbl.10",
"disk:1:\\inform3\H$\30G\supplier.tbl.11",
"disk:1:\\inform3\H$\30G\supplier.tbl.12",
"disk:1:\\inform3\H$\30G\supplier.tbl.13",
"disk:1:\\inform3\H$\30G\supplier.tbl.14",
"disk:1:\\inform3\H$\30G\supplier.tbl.15",
"disk:1:\\inform3\H$\30G\supplier.tbl.16",
"disk:1:\\inform3\H$\30G\supplier.tbl.17",
"disk:1:\\inform3\H$\30G\supplier.tbl.18",
"disk:1:\\inform3\H$\30G\supplier.tbl.19",
"disk:1:\\inform3\H$\30G\supplier.tbl.20",
"disk:1:\\inform3\I$\30G\supplier.tbl.21",
"disk:1:\\inform3\I$\30G\supplier.tbl.22",
"disk:1:\\inform3\I$\30G\supplier.tbl.23",
"disk:1:\\inform3\I$\30G\supplier.tbl.24",
"disk:1:\\inform3\I$\30G\supplier.tbl.25",
"disk:1:\\inform3\I$\30G\supplier.tbl.26",
"disk:1:\\inform3\I$\30G\supplier.tbl.27",
"disk:1:\\inform3\I$\30G\supplier.tbl.28",
"disk:1:\\inform3\I$\30G\supplier.tbl.29",
"disk:1:\\inform3\I$\30G\supplier.tbl.30",
"disk:1:\\inform3\J$\30G\supplier.tbl.31",
"disk:1:\\inform3\J$\30G\supplier.tbl.32",
"disk:1:\\inform3\J$\30G\supplier.tbl.33",
"disk:1:\\inform3\J$\30G\supplier.tbl.34",
"disk:1:\\inform3\J$\30G\supplier.tbl.35",
"disk:1:\\inform3\J$\30G\supplier.tbl.36",
"disk:1:\\inform3\J$\30G\supplier.tbl.37",
"disk:1:\\inform3\J$\30G\supplier.tbl.38",
"disk:1:\\inform3\J$\30G\supplier.tbl.39",
"disk:1:\\inform3\J$\30G\supplier.tbl.40",
"disk:1:\\inform3\K$\30G\supplier.tbl.41",
"disk:1:\\inform3\K$\30G\supplier.tbl.42",
"disk:1:\\inform3\K$\30G\supplier.tbl.43",
"disk:1:\\inform3\K$\30G\supplier.tbl.44",
"disk:1:\\inform3\K$\30G\supplier.tbl.45",
"disk:1:\\inform3\K$\30G\supplier.tbl.46",
"disk:1:\\inform3\K$\30G\supplier.tbl.47",
"disk:1:\\inform3\K$\30G\supplier.tbl.48",
"disk:1:\\inform3\K$\30G\supplier.tbl.49",
"disk:1:\\inform3\K$\30G\supplier.tbl.50",
"disk:1:\\inform3\L$\30G\supplier.tbl.51",
"disk:1:\\inform3\L$\30G\supplier.tbl.52",
"disk:1:\\inform3\L$\30G\supplier.tbl.53",
"disk:1:\\inform3\L$\30G\supplier.tbl.54",
"disk:1:\\inform3\L$\30G\supplier.tbl.55",
"disk:1:\\inform3\L$\30G\supplier.tbl.56",
"disk:1:\\inform3\L$\30G\supplier.tbl.57",
"disk:1:\\inform3\L$\30G\supplier.tbl.58",
"disk:1:\\inform3\L$\30G\supplier.tbl.59",
"disk:1:\\inform3\L$\30G\supplier.tbl.60",
"disk:1:\\inform3\M$\30G\supplier.tbl.61",
"disk:1:\\inform3\M$\30G\supplier.tbl.62",
"disk:1:\\inform3\M$\30G\supplier.tbl.63",
"disk:1:\\inform3\M$\30G\supplier.tbl.64",
"disk:1:\\inform3\M$\30G\supplier.tbl.65",
"disk:1:\\inform3\M$\30G\supplier.tbl.66",
"disk:1:\\inform3\M$\30G\supplier.tbl.67",
"disk:1:\\inform3\M$\30G\supplier.tbl.68",
"disk:1:\\inform3\M$\30G\supplier.tbl.69",
"disk:1:\\inform3\M$\30G\supplier.tbl.70",
"disk:1:\\inform3\N$\30G\supplier.tbl.71",
"disk:1:\\inform3\N$\30G\supplier.tbl.72",
"disk:1:\\inform3\N$\30G\supplier.tbl.73",
"disk:1:\\inform3\N$\30G\supplier.tbl.74",
"disk:1:\\inform3\N$\30G\supplier.tbl.75",
"disk:1:\\inform3\N$\30G\supplier.tbl.76",
"disk:1:\\inform3\N$\30G\supplier.tbl.77",
"disk:1:\\inform3\N$\30G\supplier.tbl.78",
"disk:1:\\inform3\N$\30G\supplier.tbl.79",
"disk:1:\\inform3\N$\30G\supplier.tbl.80"
),
rejectfile "\\inform3\N$\30G\supplier%c.rej",
express
);
insert into supplier select * from supp_ext;
drop table supp_ext;
```

```
alter table partsupp type (raw);
create external table partsupp_ext
sameas partsupp
using (
format "delimited",
datafiles (
"disk:1:\\inform3\G$\30G\partsupp.tbl.1",
"disk:1:\\inform3\G$\30G\partsupp.tbl.2",
"disk:1:\\inform3\G$\30G\partsupp.tbl.3",
"disk:1:\\inform3\G$\30G\partsupp.tbl.4",
"disk:1:\\inform3\G$\30G\partsupp.tbl.5",
"disk:1:\\inform3\G$\30G\partsupp.tbl.6",
"disk:1:\\inform3\G$\30G\partsupp.tbl.7",
"disk:1:\\inform3\G$\30G\partsupp.tbl.8",
"disk:1:\\inform3\G$\30G\partsupp.tbl.9",
"disk:1:\\inform3\G$\30G\partsupp.tbl.10",
"disk:1:\\inform3\H$\30G\partsupp.tbl.11",
"disk:1:\\inform3\H$\30G\partsupp.tbl.12",
"disk:1:\\inform3\H$\30G\partsupp.tbl.13",
"disk:1:\\inform3\H$\30G\partsupp.tbl.14",
"disk:1:\\inform3\H$\30G\partsupp.tbl.15",
"disk:1:\\inform3\H$\30G\partsupp.tbl.16",
"disk:1:\\inform3\H$\30G\partsupp.tbl.17",
"disk:1:\\inform3\H$\30G\partsupp.tbl.18",
"disk:1:\\inform3\H$\30G\partsupp.tbl.19",
"disk:1:\\inform3\H$\30G\partsupp.tbl.20",
"disk:1:\\inform3\I$\30G\partsupp.tbl.21",
"disk:1:\\inform3\I$\30G\partsupp.tbl.22",
"disk:1:\\inform3\I$\30G\partsupp.tbl.23",
"disk:1:\\inform3\I$\30G\partsupp.tbl.24",
"disk:1:\\inform3\I$\30G\partsupp.tbl.25",
"disk:1:\\inform3\I$\30G\partsupp.tbl.26",
"disk:1:\\inform3\I$\30G\partsupp.tbl.27",
"disk:1:\\inform3\I$\30G\partsupp.tbl.28",
"disk:1:\\inform3\I$\30G\partsupp.tbl.29",
"disk:1:\\inform3\I$\30G\partsupp.tbl.30",
"disk:1:\\inform3\J$\30G\partsupp.tbl.31",
"disk:1:\\inform3\J$\30G\partsupp.tbl.32",
"disk:1:\\inform3\J$\30G\partsupp.tbl.33",
"disk:1:\\inform3\J$\30G\partsupp.tbl.34",
"disk:1:\\inform3\J$\30G\partsupp.tbl.35",
"disk:1:\\inform3\J$\30G\partsupp.tbl.36",
"disk:1:\\inform3\J$\30G\partsupp.tbl.37",
"disk:1:\\inform3\J$\30G\partsupp.tbl.38",
"disk:1:\\inform3\J$\30G\partsupp.tbl.39",
"disk:1:\\inform3\J$\30G\partsupp.tbl.40",
"disk:1:\\inform3\K$\30G\partsupp.tbl.41",
"disk:1:\\inform3\K$\30G\partsupp.tbl.42",
"disk:1:\\inform3\K$\30G\partsupp.tbl.43",
"disk:1:\\inform3\K$\30G\partsupp.tbl.44",
"disk:1:\\inform3\K$\30G\partsupp.tbl.45",
"disk:1:\\inform3\K$\30G\partsupp.tbl.46",
"disk:1:\\inform3\K$\30G\partsupp.tbl.47",
"disk:1:\\inform3\K$\30G\partsupp.tbl.48",
"disk:1:\\inform3\K$\30G\partsupp.tbl.49",
"disk:1:\\inform3\K$\30G\partsupp.tbl.50",
"disk:1:\\inform3\L$\30G\partsupp.tbl.51",
"disk:1:\\inform3\L$\30G\partsupp.tbl.52",
"disk:1:\\inform3\L$\30G\partsupp.tbl.53",
"disk:1:\\inform3\L$\30G\partsupp.tbl.54",
"disk:1:\\inform3\L$\30G\partsupp.tbl.55",
"disk:1:\\inform3\L$\30G\partsupp.tbl.56",
"disk:1:\\inform3\L$\30G\partsupp.tbl.57",
"disk:1:\\inform3\L$\30G\partsupp.tbl.58",
"disk:1:\\inform3\L$\30G\partsupp.tbl.59",
"disk:1:\\inform3\L$\30G\partsupp.tbl.60",
"disk:1:\\inform3\M$\30G\partsupp.tbl.61",
"disk:1:\\inform3\M$\30G\partsupp.tbl.62",
"disk:1:\\inform3\M$\30G\partsupp.tbl.63",
"disk:1:\\inform3\M$\30G\partsupp.tbl.64",
"disk:1:\\inform3\M$\30G\partsupp.tbl.65",
"disk:1:\\inform3\M$\30G\partsupp.tbl.66",
"disk:1:\\inform3\M$\30G\partsupp.tbl.67",
"disk:1:\\inform3\M$\30G\partsupp.tbl.68",
"disk:1:\\inform3\M$\30G\partsupp.tbl.69",
"disk:1:\\inform3\M$\30G\partsupp.tbl.70",
"disk:1:\\inform3\N$\30G\partsupp.tbl.71",
"disk:1:\\inform3\N$\30G\partsupp.tbl.72",
"disk:1:\\inform3\N$\30G\partsupp.tbl.73",
"disk:1:\\inform3\N$\30G\partsupp.tbl.74",
"disk:1:\\inform3\N$\30G\partsupp.tbl.75",
"disk:1:\\inform3\N$\30G\partsupp.tbl.76",
"disk:1:\\inform3\N$\30G\partsupp.tbl.77",
"disk:1:\\inform3\N$\30G\partsupp.tbl.78",
"disk:1:\\inform3\N$\30G\partsupp.tbl.79",
"disk:1:\\inform3\N$\30G\partsupp.tbl.80"
),
rejectfile "\\inform3\N$\30G\partsupp%c.rej",
express
);
insert into partsupp select * from partsupp_ext;
drop table partsupp_ext;

alter table part type (raw);
create external table part_ext
sameas part
using (
format "delimited",
datafiles (
"disk:1:\\inform3\G$\30G\part.tbl.1",
"disk:1:\\inform3\G$\30G\part.tbl.2",
"disk:1:\\inform3\G$\30G\part.tbl.3",
"disk:1:\\inform3\G$\30G\part.tbl.4",
"disk:1:\\inform3\G$\30G\part.tbl.5",
"disk:1:\\inform3\G$\30G\part.tbl.6",
"disk:1:\\inform3\G$\30G\part.tbl.7",
"disk:1:\\inform3\G$\30G\part.tbl.8",
"disk:1:\\inform3\G$\30G\part.tbl.9",
"disk:1:\\inform3\G$\30G\part.tbl.10",
"disk:1:\\inform3\H$\30G\part.tbl.11",
"disk:1:\\inform3\H$\30G\part.tbl.12",
"disk:1:\\inform3\H$\30G\part.tbl.13",
"disk:1:\\inform3\H$\30G\part.tbl.14",
"disk:1:\\inform3\H$\30G\part.tbl.15",
"disk:1:\\inform3\H$\30G\part.tbl.16",
"disk:1:\\inform3\H$\30G\part.tbl.17",
"disk:1:\\inform3\H$\30G\part.tbl.18",
"disk:1:\\inform3\H$\30G\part.tbl.19",
"disk:1:\\inform3\H$\30G\part.tbl.20",
"disk:1:\\inform3\I$\30G\part.tbl.21",
"disk:1:\\inform3\I$\30G\part.tbl.22",
"disk:1:\\inform3\I$\30G\part.tbl.23",
"disk:1:\\inform3\I$\30G\part.tbl.24",
"disk:1:\\inform3\I$\30G\part.tbl.25",
"disk:1:\\inform3\I$\30G\part.tbl.26",
"disk:1:\\inform3\I$\30G\part.tbl.27",
"disk:1:\\inform3\I$\30G\part.tbl.28",
"disk:1:\\inform3\I$\30G\part.tbl.29",
"disk:1:\\inform3\I$\30G\part.tbl.30",
"disk:1:\\inform3\J$\30G\part.tbl.31",
"disk:1:\\inform3\J$\30G\part.tbl.32",
"disk:1:\\inform3\J$\30G\part.tbl.33",
"disk:1:\\inform3\J$\30G\part.tbl.34",
"disk:1:\\inform3\J$\30G\part.tbl.35",
"disk:1:\\inform3\J$\30G\part.tbl.36",
"disk:1:\\inform3\J$\30G\part.tbl.37",
"disk:1:\\inform3\J$\30G\part.tbl.38",
"disk:1:\\inform3\J$\30G\part.tbl.39",
"disk:1:\\inform3\J$\30G\part.tbl.40",
"disk:1:\\inform3\K$\30G\part.tbl.41",
"disk:1:\\inform3\K$\30G\part.tbl.42",
```

```
"disk:1:\\inform3\K$\30G\part.tbl.43",
"disk:1:\\inform3\K$\30G\part.tbl.44",
"disk:1:\\inform3\K$\30G\part.tbl.45",
"disk:1:\\inform3\K$\30G\part.tbl.46",
"disk:1:\\inform3\K$\30G\part.tbl.47",
"disk:1:\\inform3\K$\30G\part.tbl.48",
"disk:1:\\inform3\K$\30G\part.tbl.49",
"disk:1:\\inform3\K$\30G\part.tbl.50",
"disk:1:\\inform3\L$\30G\part.tbl.51",
"disk:1:\\inform3\L$\30G\part.tbl.52",
"disk:1:\\inform3\L$\30G\part.tbl.53",
"disk:1:\\inform3\L$\30G\part.tbl.54",
"disk:1:\\inform3\L$\30G\part.tbl.55",
"disk:1:\\inform3\L$\30G\part.tbl.56",
"disk:1:\\inform3\L$\30G\part.tbl.57",
"disk:1:\\inform3\L$\30G\part.tbl.58",
"disk:1:\\inform3\L$\30G\part.tbl.59",
"disk:1:\\inform3\L$\30G\part.tbl.60",
"disk:1:\\inform3\M$\30G\part.tbl.61",
"disk:1:\\inform3\M$\30G\part.tbl.62",
"disk:1:\\inform3\M$\30G\part.tbl.63",
"disk:1:\\inform3\M$\30G\part.tbl.64",
"disk:1:\\inform3\M$\30G\part.tbl.65",
"disk:1:\\inform3\M$\30G\part.tbl.66",
"disk:1:\\inform3\M$\30G\part.tbl.67",
"disk:1:\\inform3\M$\30G\part.tbl.68",
"disk:1:\\inform3\M$\30G\part.tbl.69",
"disk:1:\\inform3\M$\30G\part.tbl.70",
"disk:1:\\inform3\N$\30G\part.tbl.71",
"disk:1:\\inform3\N$\30G\part.tbl.72",
"disk:1:\\inform3\N$\30G\part.tbl.73",
"disk:1:\\inform3\N$\30G\part.tbl.74",
"disk:1:\\inform3\N$\30G\part.tbl.75",
"disk:1:\\inform3\N$\30G\part.tbl.76",
"disk:1:\\inform3\N$\30G\part.tbl.77",
"disk:1:\\inform3\N$\30G\part.tbl.78",
"disk:1:\\inform3\N$\30G\part.tbl.79",
"disk:1:\\inform3\N$\30G\part.tbl.80"
),
rejectfile "\\inform3\N$\30G\part%c.rej",
express
);
insert into part select * from part_ext;
drop table part_ext;

alter table region type (operational);
alter table nation type (operational);
alter table part type (operational);
alter table partsupp type (operational);
alter table supplier type (operational);
alter table order type (operational);
alter table customer type (operational);
alter table lineitem type (operational);
```

B-19: update_stats.sql

```
set pdqpriority high;
begin work;
update statistics medium resolution 0.1 0.95;
commit work;
```

B-20: alter_it.sql

```
alter table nation lock mode (page);
alter table region lock mode (page);
alter table supplier lock mode (page);
alter table part lock mode (page);
alter table partsupp lock mode (page);
alter table customer lock mode (page);
alter table order lock mode (page);
alter table lineitem lock mode (page);
```

B-21: create_indexes.sql

```
set pdqpriority high;
create index ocod on order(o_clerk,o_orderkey,o_orderdate)
fragment by hybrid(o_orderkey) expression
o_clerk < 'Clerk#000001071'  in o_ocod1,
o_clerk >= 'Clerk#000001071' and o_clerk <
'Clerk#000002142'  in o_ocod2,
o_clerk >= 'Clerk#000002142' and o_clerk <
'Clerk#000003213'  in o_ocod3,
o_clerk >= 'Clerk#000003213' and o_clerk <
'Clerk#000004284'  in o_ocod4,
o_clerk >= 'Clerk#000004284' and o_clerk <
'Clerk#000005355'  in o_ocod5,
o_clerk >= 'Clerk#000005355' and o_clerk <
'Clerk#000006426'  in o_ocod6,
o_clerk >= 'Clerk#000006426' and o_clerk <
'Clerk#000007497'  in o_ocod7,
o_clerk >= 'Clerk#000007497' and o_clerk <
'Clerk#000008568'  in o_ocod8,
o_clerk >= 'Clerk#000008568' and o_clerk <
'Clerk#000009639'  in o_ocod9,
o_clerk >= 'Clerk#000009639' and o_clerk <
'Clerk#000010710'  in o_ocod10,
o_clerk >= 'Clerk#000010710' and o_clerk <
'Clerk#000011781'  in o_ocod11,
o_clerk >= 'Clerk#000011781' and o_clerk <
'Clerk#000012852'  in o_ocod12,
o_clerk >= 'Clerk#000012852' and o_clerk <
'Clerk#000013923'  in o_ocod13,
o_clerk >= 'Clerk#000013923' and o_clerk <
'Clerk#000014994'  in o_ocod14,
o_clerk >= 'Clerk#000014994' and o_clerk <
'Clerk#000016065'  in o_ocod15,
o_clerk >= 'Clerk#000016065' and o_clerk <
'Clerk#000017136'  in o_ocod16,
o_clerk >= 'Clerk#000017136' and o_clerk <
'Clerk#000018207'  in o_ocod17,
o_clerk >= 'Clerk#000018207' and o_clerk <
'Clerk#000019278'  in o_ocod18,
o_clerk >= 'Clerk#000019278' and o_clerk <
'Clerk#000020349'  in o_ocod19,
o_clerk >= 'Clerk#000020349' and o_clerk <
'Clerk#000021420'  in o_ocod20,
o_clerk >= 'Clerk#000021420' and o_clerk <
'Clerk#000022491'  in o_ocod21,
o_clerk >= 'Clerk#000022491' and o_clerk <
'Clerk#000023562'  in o_ocod22,
o_clerk >= 'Clerk#000023562' and o_clerk <
'Clerk#000024633'  in o_ocod23,
o_clerk >= 'Clerk#000024633' and o_clerk <
'Clerk#000025704'  in o_ocod24,
o_clerk >= 'Clerk#000025704' and o_clerk <
'Clerk#000026775'  in o_ocod25,
o_clerk >= 'Clerk#000026775' and o_clerk <
'Clerk#000027846'  in o_ocod26,
o_clerk >= 'Clerk#000027846' and o_clerk <
'Clerk#000028917'  in o_ocod27,
o_clerk >= 'Clerk#000028917'  in o_ocod28;

create index lored on
lineitem(l_orderkey,l_returnflag,l_extendedprice,l_discount)
fragment by hash(l_orderkey) in l_lored;

create index lpqesod on
lineitem(l_partkey,l_quantity,l_extendedprice)
fragment by hybrid(l_orderkey) expression
```

```
        l_partkey < 214280  in l_pqesod1,
        l_partkey >= 214280 and l_partkey < 428560  in
l_pqesod2,
        l_partkey >= 428560 and l_partkey < 642840  in
l_pqesod3,
        l_partkey >= 642840 and l_partkey < 857120  in
l_pqesod4,
        l_partkey >= 857120 and l_partkey < 1071400  in
l_pqesod5,
        l_partkey >= 1071400 and l_partkey < 1285680  in
l_pqesod6,
        l_partkey >= 1285680 and l_partkey < 1499960  in
l_pqesod7,
        l_partkey >= 1499960 and l_partkey < 1714240  in
l_pqesod8,
        l_partkey >= 1714240 and l_partkey < 1928520  in
l_pqesod9,
        l_partkey >= 1928520 and l_partkey < 2142800  in
l_pqesod10,
        l_partkey >= 2142800 and l_partkey < 2357080  in
l_pqesod11,
        l_partkey >= 2357080 and l_partkey < 2571360  in
l_pqesod12,
        l_partkey >= 2571360 and l_partkey < 2785640  in
l_pqesod13,
        l_partkey >= 2785640 and l_partkey < 2999920  in
l_pqesod14,
        l_partkey >= 2999920 and l_partkey < 3214200  in
l_pqesod15,
        l_partkey >= 3214200 and l_partkey < 3428400  in
l_pqesod16,
        l_partkey >= 3428400 and l_partkey < 3642760  in
l_pqesod17,
        l_partkey >= 3642760 and l_partkey < 3857040  in
l_pqesod18,
        l_partkey >= 3857040 and l_partkey < 4071320  in
l_pqesod19,
        l_partkey >= 4071320 and l_partkey < 4285600  in
l_pqesod20,
        l_partkey >= 4285600 and l_partkey < 4499880  in
l_pqesod21,
        l_partkey >= 4499880 and l_partkey < 4714160  in
l_pqesod22,
        l_partkey >= 4714160 and l_partkey < 4928440  in
l_pqesod23,
        l_partkey >= 4928440 and l_partkey < 5142720  in
l_pqesod24,
        l_partkey >= 5142720 and l_partkey < 5357000  in
l_pqesod25,
        l_partkey >= 5357000 and l_partkey < 5571280  in
l_pqesod26,
        l_partkey >= 5571280 and l_partkey < 5785560  in
l_pqesod27,
        l_partkey >= 5785560 in l_pqesod28;

create unique index pspss on
partsupp(ps_partkey,ps_suppkey,ps_supplycost)
fragment by hybrid(ps_partkey) expression
        ps_partkey < 214280  in ps_pspss1,
        ps_partkey >= 214280 and ps_partkey < 428560  in
ps_pspss2,
        ps_partkey >= 428560 and ps_partkey < 642840  in
ps_pspss3,
        ps_partkey >= 642840 and ps_partkey < 857120  in
ps_pspss4,
        ps_partkey >= 857120 and ps_partkey < 1071400  in
ps_pspss5,
        ps_partkey >= 1071400 and ps_partkey < 1285680  in
ps_pspss6,
        ps_partkey >= 1285680 and ps_partkey < 1499960  in
ps_pspss7,
        ps_partkey >= 1499960 and ps_partkey < 1714240  in
ps_pspss8,
        ps_partkey >= 1714240 and ps_partkey < 1928520  in
ps_pspss9,
        ps_partkey >= 1928520 and ps_partkey < 2142800  in
ps_pspss10,
        ps_partkey >= 2142800 and ps_partkey < 2357080  in
ps_pspss11,
        ps_partkey >= 2357080 and ps_partkey < 2571360  in
ps_pspss12,
        ps_partkey >= 2571360 and ps_partkey < 2785640  in
ps_pspss13,
        ps_partkey >= 2785640 and ps_partkey < 2999920  in
ps_pspss14,
        ps_partkey >= 2999920 and ps_partkey < 3214200  in
ps_pspss15,
        ps_partkey >= 3214200 and ps_partkey < 3428400  in
ps_pspss16,
        ps_partkey >= 3428400 and ps_partkey < 3642760  in
ps_pspss17,
        ps_partkey >= 3642760 and ps_partkey < 3857040  in
ps_pspss18,
        ps_partkey >= 3857040 and ps_partkey < 4071320  in
ps_pspss19,
        ps_partkey >= 4071320 and ps_partkey < 4285600  in
ps_pspss20,
        ps_partkey >= 4285600 and ps_partkey < 4499880  in
ps_pspss21,
        ps_partkey >= 4499880 and ps_partkey < 4714160  in
ps_pspss22,
        ps_partkey >= 4714160 and ps_partkey < 4928440  in
ps_pspss23,
        ps_partkey >= 4928440 and ps_partkey < 5142720  in
ps_pspss24,
        ps_partkey >= 5142720 and ps_partkey < 5357000  in
ps_pspss25,
        ps_partkey >= 5357000 and ps_partkey < 5571280  in
ps_pspss26,
        ps_partkey >= 5571280 and ps_partkey < 5785560  in
ps_pspss27,
        ps_partkey >= 5785560 in ps_pspss28;

create index psspsa on
partsupp(ps_suppkey,ps_partkey,ps_supplycost,
ps_availqty)
fragment by hybrid(ps_partkey) expression
        ps_suppkey < 10710  in ps_spsa1,
        ps_suppkey >= 10710 and ps_suppkey < 21420  in
ps_spsa2,
        ps_suppkey >= 21420 and ps_suppkey < 32130  in
ps_spsa3,
        ps_suppkey >= 32130 and ps_suppkey < 42840  in
ps_spsa4,
        ps_suppkey >= 42840 and ps_suppkey < 53550  in
ps_spsa5,
        ps_suppkey >= 53550 and ps_suppkey < 64260  in
ps_spsa6,
        ps_suppkey >= 64260 and ps_suppkey < 74970  in
ps_spsa7,
        ps_suppkey >= 74970 and ps_suppkey < 85680  in
ps_spsa8,
        ps_suppkey >= 85680 and ps_suppkey < 96390  in
ps_spsa9,
        ps_suppkey >= 96390 and ps_suppkey < 107100  in
ps_spsa10,
        ps_suppkey >= 107100 and ps_suppkey < 117810  in
ps_spsa11,
        ps_suppkey >= 117810 and ps_suppkey < 128520  in
ps_spsa12,
        ps_suppkey >= 128520 and ps_suppkey < 139230  in
ps_spsa13,
```

```
        ps_suppkey >= 139230 and ps_suppkey < 149940  in
ps_spsa14,
        ps_suppkey >= 149940 and ps_suppkey < 160650  in
ps_spsa15,
        ps_suppkey >= 160650 and ps_suppkey < 171360  in
ps_spsa16,
        ps_suppkey >= 171360 and ps_suppkey < 182070  in
ps_spsa17,
        ps_suppkey >= 182070 and ps_suppkey < 192780  in
ps_spsa18,
        ps_suppkey >= 192780 and ps_suppkey < 203490  in
ps_spsa19,
        ps_suppkey >= 203490 and ps_suppkey < 214200  in
ps_spsa20,
        ps_suppkey >= 214200 and ps_suppkey < 224910  in
ps_spsa21,
        ps_suppkey >= 224910 and ps_suppkey < 235620  in
ps_spsa22,
        ps_suppkey >= 235620 and ps_suppkey < 246330  in
ps_spsa23,
        ps_suppkey >= 246330 and ps_suppkey < 257040  in
ps_spsa24,
        ps_suppkey >= 257040 and ps_suppkey < 267750  in
ps_spsa25,
        ps_suppkey >= 267750 and ps_suppkey < 278460  in
ps_spsa26,
        ps_suppkey >= 278460 and ps_suppkey < 289170  in
ps_spsa27,
        ps_suppkey >= 289170  in ps_spsa28;

create index ookey on
order(o_orderkey)
fragment by hash(o_orderkey) in o_okey1;
```

Appendix C: Query Validation EQT and Output

C-1: Query 1

```
-- QUERY 1 PRICING SUMMARY REPORT QUERY
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 <= date('1998-12-01') - interval (90) day
(3) to day
group by l_returnflag,l_linestatus
order by l_returnflag,l_linestatus;


l_returnflag    A
l_linestatus    F
sum_qty         3773034.00
sum_base_price  5319329289.68
sum_disc_price  5053976845.7839
sum_charge      5256336547.68
avg_qty         25.51
avg_price       35964.01
avg_disc        0.05
count_order     147907

l_returnflag    N
l_linestatus    F
sum_qty         100245.00
sum_base_price  141459686.10
sum_disc_price  134380852.7691
sum_charge      139710306.87
avg_qty         25.63
avg_price       36160.45
avg_disc        0.05
count_order     3912

l_returnflag    N
l_linestatus    O
sum_qty         7464940.00
sum_base_price  10518546073.98
sum_disc_price  9992072944.4612
sum_charge      10392414192.06
avg_qty         25.54
avg_price       35990.13
avg_disc        0.05
count_order     292262

l_returnflag    R
l_linestatus    F
sum_qty         3779140.00
sum_base_price  5328886172.99
sum_disc_price  5062370635.9343
sum_charge      5265431221.82
avg_qty         25.55
avg_price       36025.46
avg_disc        0.05
count_order     147920

4 row(s) retrieved.


commit work;
Data committed.

Query: 1 Date: 1997-07-22 Time: 11:44:10.765  336880.765


begin work;
Started transaction.


-- using default substitutions
```

C-2: Query 2

```
--  QUERY 2 MINIMUM COST SUPPLIER QUERY
select
FIRST 100
        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;


s_acctbal  9828.21
s_name     Supplier#000000647
n_name     UNITED KINGDOM
p_partkey  13120
p_mfgr     Manufacturer#5
s_address  jB16PyPyB7Bl52jMjSPw3mS
s_phone    33-258-202-4782
s_comment  z1QhSiMj11Bm7COlLwh6Ql0B1R2Mg4CLn
LhiP0wiMzy72hlkP7l5in2y6RS6N130lz
        51nSRL5gOg5S26hPCCQN2L

s_acctbal  9508.37
s_name     Supplier#000000070
```

n_name FRANCE
p_partkey 3563
p_mfgr Manufacturer#1
s_address M5C616R5h5SlMR3zzmLkSw24j2
s_phone 16-821-608-1166
s_comment m7z0CPShmBkhlChBAi3LkQ2CLw
mhl6QP362RPS3044CB2y41yhOhjlBin0CL7yhxmhS
4hBM07kQ1yyjOjz3C

s_acctbal 9508.37
s_name Supplier#000000070
n_name FRANCE
p_partkey 17268
p_mfgr Manufacturer#4
s_address M5C616R5h5SlMR3zzmLkSw24j2
s_phone 16-821-608-1166
s_comment m7z0CPShmBkhlChBAi3LkQ2CLw
mhl6QP362RPS3044CB2y41yhOhjlBin0CL7yhxmhS
4hBM07kQ1yyjOjz3C

s_acctbal 9453.01
s_name Supplier#000000802
n_name ROMANIA
p_partkey 10021
p_mfgr Manufacturer#5
s_address
5yARQNSLNRAlOlBnkNQCik3SOlyClk7nmRhA2h0
s_phone 29-342-882-6463
s_comment 65y3RQ2i0OP6Nz7mS hC
PxwLy7L1jQy6Ol63xO3iBCz52Rm1zm0MziCMLij2n6wky5
1
mBOwx Qh52iz QB1545Amxyj

s_acctbal 9453.01
s_name Supplier#000000802
n_name ROMANIA
p_partkey 13275
p_mfgr Manufacturer#4
s_address
5yARQNSLNRAlOlBnkNQCik3SOlyClk7nmRhA2h0
s_phone 29-342-882-6463
s_comment 65y3RQ2i0OP6Nz7mS hC
PxwLy7L1jQy6Ol63xO3iBCz52Rm1zm0MziCMLij2n6wky5
1
mBOwx Qh52iz QB1545Amxyj

s_acctbal 9192.10
s_name Supplier#000000115
n_name UNITED KINGDOM
p_partkey 13325
p_mfgr Manufacturer#1
s_address h0m3lzlSPMw2B0ny7LNyNCkjRRn7iyMlLBLA
s_phone 33-597-248-1220
s_comment 1QzQjhSyx
ixm2lgz2Ry7075RL3MS5z36x56hxmR0wLN0LBxm164LzC
MmALzOAjn4kz7
i4wjOlCONl1C51M7nCMx66SBRAQA

s_acctbal 9032.15
s_name Supplier#000000959
n_name GERMANY
p_partkey 4958
p_mfgr Manufacturer#4
s_address 205LNCzxMCnQ5gnz4n S3ynP6Mhnw
s_phone 17-108-642-3106
s_comment Px z7kOx56l7jQz NwBBQhky
yM7kLgxRQw5zw6 426Bm55lC6 OkQ7hQPLixjM7y47B
NP16CRi0kjk354lgxh

s_acctbal 8702.02
s_name Supplier#000000333
n_name RUSSIA
p_partkey 11810
p_mfgr Manufacturer#3
s_address 5iwkgN5n2BN15OmQk2602h0N6NzxPyiPN5lnj
s_phone 32-508-202-6136
s_comment SgimAjmn3wL7Rlxmh3LCwOPnhjyl 7xxzxAN
4ACx43y65NwQ7P

s_acctbal 8615.50
s_name Supplier#000000812
n_name FRANCE
p_partkey 10551
p_mfgr Manufacturer#2
s_address h4i2M2O0 ky1g2mlBOmxjzj0hA2h6nkSNhP
s_phone 16-585-724-6633
s_comment 57i0NAyR0RP2jOh54C6B22OlSL

s_acctbal 8615.50
s_name Supplier#000000812
n_name FRANCE
p_partkey 13811
p_mfgr Manufacturer#4
s_address h4i2M2O0 ky1g2mlBOmxjzj0hA2h6nkSNhP
s_phone 16-585-724-6633
s_comment 57i0NAyR0RP2jOh54C6B22OlSL

s_acctbal 8488.53
s_name Supplier#000000367
n_name RUSSIA
p_partkey 6854
p_mfgr Manufacturer#4
s_address nkmQ2Qzgh0wA 3x Sn2S7N5gmSOj xwC COSn6
s_phone 32-458-198-9557
s_comment 35C2RROP C Nlgi2N
SxAj0hQkn7kP5z4wSxSwgMxj6k4MRmh0S2Qm7R3z4jB
OOQBM
l

s_acctbal 8430.52
s_name Supplier#000000646
n_name FRANCE
p_partkey 11384
p_mfgr Manufacturer#3
s_address 61SjP6S y B0 32l11
s_phone 16-601-220-5489
s_comment
kiw4NSNBNxy5kywzwyx0PMM21xiMOhxR423Akkm
Q7CNwRzQS23Nzz22 mnm6P377Q3M
j7n 56BLm6lxwllh kSmN

s_acctbal 8271.39
s_name Supplier#000000146
n_name RUSSIA
p_partkey 4637
p_mfgr Manufacturer#5
s_address wh yPSk6hNBlB4l33iQ0wS0 RhBhQ4zQ3lz
s_phone 32-792-619-3155
s_comment jjwgwljRO63
n7OM2MP0hg3L1mlwBMLmMlS4Cgyn
LA5PwC2P0AS6g3C5mkOjO72NPig
731m

s_acctbal 8096.98
s_name Supplier#000000574
n_name RUSSIA
p_partkey 323
p_mfgr Manufacturer#4
s_address hCOj4Cgx43xx jgP4QkL7gLN65
s_phone 32-866-246-8752
s_comment OhxNj6SlB56315B3k5SCBzwQyLk76zlj4Ow2Q
BC2wACkxh3S0RCyx6nARzSQR2010k0

BCPhOg6yQm

s_acctbal 7392.78
s_name Supplier#000000170
n_name UNITED KINGDOM
p_partkey 7655
p_mfgr Manufacturer#2
s_address PCxjjzNQihLNxgLw0SiMmQ
s_phone 33-803-340-5398
s_comment M116S1xzg54iC3k7OPLQi3Cimhghz2BClQk
g5Ag12QSBlhglANnw4MR MBS 72A

s_acctbal 7205.20
s_name Supplier#000000477
n_name GERMANY
p_partkey 10956
p_mfgr Manufacturer#5
s_address Mimj6403h zmAzAgg Bjy05O 2z
s_phone 17-180-144-7991
s_comment yRlyR SnMxmhPjAmBw
S02AxQ6yOhBRlOwzmlxzO0A2Sx075kjlAknn7z2
O0S7hy0Bi
knwOQm6Pmz3gL4gj2z7

s_acctbal 6820.35
s_name Supplier#000000007
n_name UNITED KINGDOM
p_partkey 13217
p_mfgr Manufacturer#5
s_address z45m2jBRzl5ilLNz4
s_phone 33-990-965-2201
s_comment 1PhngjmiSQl0RzRACP0l4S70xSL
QPSBM16072SkMLCgm4OOMjARLNQk3g1P3BB32AgB
Ml462B0CP7Rh24

s_acctbal 6721.70
s_name Supplier#000000954
n_name FRANCE
p_partkey 4191
p_mfgr Manufacturer#3
s_address OM7xnNxNnkgQ mzh2g3RQmg1g
s_phone 16-537-341-8517
s_comment 5ni3yCkmz5ymx0kCg74zhLA
B516Si1w152AkiByx1Nl NgghAkkmNz1jASj4mxzxznO
ySg7hAyM3MRRnBj

s_acctbal 6329.90
s_name Supplier#000000996
n_name GERMANY
p_partkey 10735
p_mfgr Manufacturer#2
s_address
k6135gA3zPwNl7L3Rl45mlnACjngOQQBB300iyA
s_phone 17-447-811-3282
s_comment PBO7wjlQMm1h3AAA 1NQAl0kkijnkRNgQ0
mh1z6QS0gC5lP1 ykmzNR20OlN506ARS0
z3j

s_acctbal 6173.87
s_name Supplier#000000408
n_name RUSSIA
p_partkey 18139
p_mfgr Manufacturer#1
s_address Cni6 zR5C4lh104POx5h05
mg53CQ2Sw4SAM2M2x
s_phone 32-858-724-2950
s_comment
10SxMOwhjON3khzQ124gNnyw7B4nL7ml4L5llSR

s_acctbal 5364.99
s_name Supplier#000000785
n_name RUSSIA
p_partkey 13784
p_mfgr Manufacturer#4
s_address 71OnPzQkC2P1hRNRgjyQP4n1
s_phone 32-297-653-2203
s_comment kiiPQ3ik7R ykAhRx43Rw70L1Ok
7AMi3AjRw7lklwxwyiL6S2O1COyS4QB46m5M167m
jMwCm0w

s_acctbal 5069.27
s_name Supplier#000000328
n_name GERMANY
p_partkey 16327
p_mfgr Manufacturer#1
s_address 5O4033xSgml
s_phone 17-231-513-5721
s_comment OMk3ALAPNmj6lBLMAS7M1nCAS
4xLj51iy2klix3nPi26gAxPgANmk6zSi6 3A7m l1
1BOwiC6xLB4hBRiPM

s_acctbal 4941.88
s_name Supplier#000000321
n_name ROMANIA
p_partkey 7320
p_mfgr Manufacturer#5
s_address hyLQ mg42S2kAMlj M3BwMSjS
s_phone 29-573-279-1406
s_comment
y2644kMhOkPCm5P5y7Lmz7OR6mgSmBN631RggmC

s_acctbal 4672.25
s_name Supplier#000000239
n_name RUSSIA
p_partkey 12238
p_mfgr Manufacturer#1
s_address y4ymj7B5BN1nMSkwPPggAl
s_phone 32-396-654-6826
s_comment Py3RA2gykmSCmj0z3ii7Rxzhz6OyR RxS
C3S23LPQ

s_acctbal 4586.49
s_name Supplier#000000680
n_name RUSSIA
p_partkey 5679
p_mfgr Manufacturer#3
s_address BP1Nlw5nPMxRnOAwm
s_phone 32-522-382-1620
s_comment kA0y25RNO1Al
im7SyiPzSym3M5OS52l6S576kn0S2k
0mPBLlAzL6Ax7CM6iNi4CgCy
6Bln7hlhxmlRng

s_acctbal 4518.31
s_name Supplier#000000149
n_name FRANCE
p_partkey 18344
p_mfgr Manufacturer#5
s_address 4B QSy5Bl2
s_phone 16-660-553-2456
s_comment
hijkPhgL1g4L1Q27y0Q42wh0Qz3jPiL4NgkM4NNg1
llQ1yNNBk1C1QnlRO7 4ki

s_acctbal 4315.15
s_name Supplier#000000509
n_name FRANCE
p_partkey 18972
p_mfgr Manufacturer#2
s_address B5 iPRn7L4yMllgwCnRPMA
s_phone 16-298-154-3365
s_comment ygiPh7ymP7jBznmR2lQLLgjmi1wik

s_acctbal 3526.53
s_name Supplier#000000553
n_name FRANCE
p_partkey 8036
p_mfgr Manufacturer#4
s_address ylLOx2gMw 5iBl6AiNL60Q
s_phone 16-599-552-3755
s_comment L3ggShlRlyxmR4MNl7Rw7OQign6yO

s_acctbal 3526.53
s_name Supplier#000000553
n_name FRANCE
p_partkey 17018
p_mfgr Manufacturer#3
s_address ylLOx2gMw 5iBl6AiNL60Q
s_phone 16-599-552-3755
s_comment L3ggShlRlyxmR4MNl7Rw7OQign6yO

s_acctbal 3294.68
s_name Supplier#000000350
n_name GERMANY
p_partkey 4841
p_mfgr Manufacturer#4
s_address x5kRL2z1BPg0 BO 2hi1iOyh 30RRg0OPj
s_phone 17-113-181-4017
s_comment BjQznni44OmQ7S16yl3zxk2M6nM4M
27yMPML

s_acctbal 2972.26
s_name Supplier#000000016
n_name RUSSIA
p_partkey 1015
p_mfgr Manufacturer#4
s_address B7wLkSLRjNS MS1C
s_phone 32-822-502-4215
s_comment C7w6S6QzhAPQmMmNmMN1hA0llQOA
00m1NmC25wyQ461SA jy03zmRh22MLM00zhmi

s_acctbal 2963.09
s_name Supplier#000000840
n_name ROMANIA
p_partkey 3080
p_mfgr Manufacturer#2
s_address 1ynwiQkNh0
CMRRCk4l306M2ij0jykg6QNgSCAzy
s_phone 29-781-337-5584
s_comment
S7NRMx43RmOjxML6hxLyN75LzxBwB0wjSLx3
S3Cwh52S6ilSOLhQm0 6C1 yzx3jPm6
Sjg 5By0BCPwOR32i1CQgxR0gB43gh

s_acctbal 2221.25
s_name Supplier#000000771
n_name ROMANIA
p_partkey 13981
p_mfgr Manufacturer#2
s_address LAjCRjl3nAMzzhmw0Sx1Mg
s_phone 29-986-304-9006
s_comment jhk0N7NlhS23iCngC52BBC
0jilM0wByx0LB5R070R2lCx113lQiS7xNhBRA0xknlNxL
iA

s_acctbal 1381.97
s_name Supplier#000000104
n_name FRANCE
p_partkey 18103
p_mfgr Manufacturer#3
s_address i Qnl4 1 jiwM C2yxAylL5R4SBQh54N6
s_phone 16-434-972-6922
s_comment
MwnBw1g71Pig2Am7nz0Mm5SNl7OwQLAkN56ji

s_acctbal 906.07
s_name Supplier#000000138
n_name ROMANIA
p_partkey 8363
p_mfgr Manufacturer#4
s_address 1iBxSxLl1Mh3 6LS6PlLPNlnlMjCQh22z6n5
s_phone 29-533-434-6776
s_comment nLjQAmCw77R2jRMgz5LSyyx1QN l
4jMMO3RAkxOkzRmwQl3Qm5236k72RRPnim0 Bkz
QnBMM6A PMml2n

s_acctbal 765.69
s_name Supplier#000000799
n_name RUSSIA
p_partkey 11276
p_mfgr Manufacturer#2
s_address Am7yihz47mg NkgQL w By4
s_phone 32-579-339-1495
s_comment MMRPNQ
4l66mQQPNniAiiL0PQ2C4yyBRn1nRlxnxkj5Ak45Pw
mQk1ROhz66BRQiiL g
PRQRy 56MyQ nS1N14R 7Ml6xhl2lOS3

s_acctbal 727.89
s_name Supplier#000000470
n_name ROMANIA
p_partkey 6213
p_mfgr Manufacturer#3
s_address gAySBM2N7 PgwP5kiP4n7BzOik0M
s_phone 29-165-289-1523
s_comment zCkPgn
6wN5A3R47gljlQ3hNSLShP2RALxCiinkOy4wCwA1LCiBO
5yiSC yBAA lii

s_acctbal 683.07
s_name Supplier#000000651
n_name RUSSIA
p_partkey 4888
p_mfgr Manufacturer#4
s_address ymQ6PByCh4lzxBBPLB2wwOhRh47wQMQSPL
s_phone 32-181-426-4490
s_comment
kx6jhQkwzg6RkRgPLPM30BgL1R726l1m5AMk0MmMQB
Q nCihiLxhMgCgRih6MmMx0Pgl
RQ7AQnl72g50

C-3: Query 3

```
-- QUERY 3 SHIPPING PRIORITY QUERY
select
FIRST 10
        l_orderkey,
        sum(l_extendedprice * (1 - l_discount)) as revenue,
        o_orderdate,
        o_shippriority
from
        customer, order, lineitem
where
        c_mktsegment = 'BUILDING' and
        c_custkey = o_custkey and
        l_orderkey = o_orderkey and
        o_orderdate < date('1995-03-15') and
        l_shipdate > date('1995-03-15')
group by l_orderkey,o_orderdate,o_shippriority
order by revenue desc,o_orderdate;

 l_orderkey          revenue o_orderdate o_shippriority
```

```
 260930    320547.2525 1995-03-12          0
 402497    298879.5320 1995-02-12          0
 457859    296490.6754 1995-01-17          0
 509889    294068.8736 1995-02-03          0
  58117   292632.8325 1995-02-21         0
 538311    279665.9960 1995-03-07          0
 588421    275477.1172 1995-03-03          0
 416167    273765.4530 1995-02-22          0
  97830   273227.0610 1995-03-04         0
  90276   272233.9174 1995-03-04         0

10 row(s) retrieved.


commit work;
Data committed.

Query: 3 Date: 1997-07-22 Time: 11:44:52.125  336922.125


begin work;
Started transaction.


-- using default substitutions
```

C-4: Query 4

```
-- QUERY 4 ORDER PRIORITY CHECKING QUERY
set explain on;
Explain set.


select
        o_orderpriority,
        count(*) as order_count
from order
where
        o_orderdate >= date('1993-07-01') and
        o_orderdate < date('1993-07-01') + interval (3)
month to month 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                999
2-HIGH                 1002
3-MEDIUM               1021
4-NOT SPECIFIED         997
5-LOW                  1089

5 row(s) retrieved.


commit work;
Data committed.

Query: 4 Date: 1997-07-22 Time: 11:44:12.437  336882.437


begin work;
Started transaction.


-- using default substitutions
```

C-5: Query 5

```
-- QUERY 5 LOCAL SUPPLIER VOLUME QUERY
select
        n_name,
        sum(l_extendedprice * ( 1 - l_discount)) as revenue
from
        customer, order, lineitem, supplier, nation, region
where
        c_custkey = o_custkey and
        o_orderkey = l_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 >= date('1994-01-01') and
        o_orderdate < date('1994-01-01') + interval (1)
year to year
group by n_name
order by revenue desc;

n_name                  revenue

CHINA                 7349391.4710
INDONESIA             6485853.4033
INDIA                5505346.8197
JAPAN                 5388883.5941
VIETNAM               4728846.6018

5 row(s) retrieved.


commit work;
Data committed.

Query: 5 Date: 1997-07-22 Time: 11:44:55.421  336925.421


begin work;
Started transaction.


-- using default substitutions
```

C-6: Query 6

```
-- QUERY 6 FORECASTING REVENUE CHANGE
QUERY
select
        sum(l_extendedprice * l_discount) as revenue
from
        lineitem
where
        l_shipdate >= date('1994-01-01') and
        l_shipdate < date('1994-01-01') + interval (1) year
to year and
        l_discount between .06 - 0.01 and .06 + 0.01 and
        l_quantity <  24;

      revenue

  11450588.0434
```

1 row(s) retrieved.

commit work;
Data committed.

Query: 6 Date: 1997-07-22 Time: 11:44:23.406 336893.406

begin work;
Started transaction.

-- using default substitutions

C-7: Query 7

```
-- QUERY 7 VOLUME SHIPPING QUERY
select
        n1.n_name as supp_nation,
        n2.n_name as cust_nation,
        year(l_shipdate) as year,
        sum(l_extendedprice * (1 - l_discount)) as revenue
from
        supplier, lineitem, order, 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 date('1995-01-01') and date('1996-12-
31')
group by n1.n_name,n2.n_name,3
order by n1.n_name,n2.n_name,3;
```

supp_nation	cust_nation	year	revenue
FRANCE	GERMANY	1995	4611421.4400
FRANCE	GERMANY	1996	4828420.3721
GERMANY	FRANCE	1995	6755766.8409
GERMANY	FRANCE	1996	5810951.3958

4 row(s) retrieved.

commit work;
Data committed.

Query: 7 Date: 1997-07-22 Time: 11:45:00.687 336930.687

begin work;
Started transaction.

-- using default substitutions

C-8: Query 8

```
-- QUERY 8 NATIONAL MARKET SHARE QUERY
select
        year(o_orderdate) as year,
        round(sum(case when n2.n_name= 'BRAZIL'
                then (l_extendedprice * (1 -
l_discount))
                else 0
        end)/sum(l_extendedprice * (1 - l_discount)),2) as
mkt_share
from
        part, supplier, lineitem, order, 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 date('1995-01-01') and
        date('1996-12-31') and
        p_type = 'ECONOMY ANODIZED STEEL'
group by 1
order by 1;
```

year	mkt_share
1995	0.05
1996	0.09

2 row(s) retrieved.

commit work;
Data committed.

Query: 8 Date: 1997-07-22 Time: 11:44:35.968 336905.968

begin work;
Started transaction.

-- using default substitutions

C-9: Query 9

```
-- QUERY 9 PRODUCT TYPE PROFIT MEASURE QUERY
select
        n_name as nation,
        year(o_orderdate) as year,
        sum(l_extendedprice *
    (1 - l_discount) - ps_supplycost * l_quantity) as
sum_profit
from
        part, supplier, lineitem, partsupp, order, 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%'
group by n_name,2
order by n_name,2 desc;
```

nation	year	sum_profit
ALGERIA	1998	1946316.0053
ALGERIA	1997	2973825.6921
ALGERIA	1996	3308881.5165
ALGERIA	1995	3092227.2988
ALGERIA	1994	3406958.7104
ALGERIA	1993	3140744.0263
ALGERIA	1992	3330704.4066
ARGENTINA	1998	3045410.0081
ARGENTINA	1997	4255378.5927
ARGENTINA	1996	4651751.9367
ARGENTINA	1995	4897797.0030
ARGENTINA	1994	4823465.7691
ARGENTINA	1993	4499810.7131
ARGENTINA	1992	4764593.3861
BRAZIL	1998	2932051.3632
BRAZIL	1997	3784531.3499
BRAZIL	1996	3965665.6899
BRAZIL	1995	4063060.8607
BRAZIL	1994	4236277.3501
BRAZIL	1993	4363461.3131
BRAZIL	1992	4684749.2328
CANADA	1998	2217064.0383
CANADA	1997	2950110.6103
CANADA	1996	3184049.9686
CANADA	1995	3962540.1948
CANADA	1994	3365251.0225
CANADA	1993	3617013.3667
CANADA	1992	3407955.2491
CHINA	1998	3048192.0230
CHINA	1997	5001207.6910
CHINA	1996	4800958.3133
CHINA	1995	5154927.7284
CHINA	1994	5882634.5341
CHINA	1993	4733364.8206
CHINA	1992	5014704.0793
EGYPT	1998	1892538.7444
EGYPT	1997	3849220.0749
EGYPT	1996	3418656.5535
EGYPT	1995	3766170.6034
EGYPT	1994	3520025.5593
EGYPT	1993	4375424.7450
EGYPT	1992	4586034.3943
ETHIOPIA	1998	1860117.7283
ETHIOPIA	1997	3705722.3335
ETHIOPIA	1996	3577215.3925
ETHIOPIA	1995	3425219.5519
ETHIOPIA	1994	3428616.1848
ETHIOPIA	1993	3459815.4314
ETHIOPIA	1992	3280072.9080
FRANCE	1998	1592531.5484
FRANCE	1997	2746176.5385
FRANCE	1996	2505844.8797
FRANCE	1995	2902077.0045
FRANCE	1994	2532229.5603
FRANCE	1993	2305725.4424
FRANCE	1992	2955126.6886
GERMANY	1998	3538625.7338
GERMANY	1997	4425943.3995
GERMANY	1996	4266344.9555
GERMANY	1995	3952963.5162
GERMANY	1994	4462655.7983
GERMANY	1993	4435094.6575
GERMANY	1992	4521715.4116
INDIA	1998	3378369.3369
INDIA	1997	4186477.8481
INDIA	1996	5074383.9250
INDIA	1995	4487435.3793
INDIA	1994	4718312.6259
INDIA	1993	4499573.8099
INDIA	1992	4712930.3331
INDONESIA	1998	2902077.1015
INDONESIA	1997	4973644.2283
INDONESIA	1996	4977652.4887
INDONESIA	1995	5359380.1510
INDONESIA	1994	4854637.1996
INDONESIA	1993	4213131.4235
INDONESIA	1992	4999478.5062
IRAN	1998	2415763.1012
IRAN	1997	4227175.1094
IRAN	1996	4527365.0271
IRAN	1995	4139514.7174
IRAN	1994	4166316.3907
IRAN	1993	3366959.5882
IRAN	1992	3599399.7018
IRAQ	1998	2596922.6334
IRAQ	1997	3707054.1118
IRAQ	1996	3726138.3835
IRAQ	1995	4350503.8921
IRAQ	1994	4131512.7911
IRAQ	1993	3787196.4208
IRAQ	1992	4043738.1336
JAPAN	1998	2265666.9424
JAPAN	1997	3988819.2811
JAPAN	1996	4319004.5339
JAPAN	1995	4262698.6369
JAPAN	1994	3545212.6196
JAPAN	1993	4051565.9746
JAPAN	1992	3692137.4454
JORDAN	1998	1978591.7418
JORDAN	1997	3315454.2870
JORDAN	1996	3236531.9798
JORDAN	1995	2778207.9861
JORDAN	1994	2420301.0715
JORDAN	1993	3272130.9349
JORDAN	1992	2649126.0864
KENYA	1998	2265677.7268
KENYA	1997	3493019.3230
KENYA	1996	3346373.2964
KENYA	1995	3537360.3249
KENYA	1994	2800950.7159
KENYA	1993	3477468.3019
KENYA	1992	2719618.0405
MOROCCO	1998	2549499.9295
MOROCCO	1997	3891824.8983
MOROCCO	1996	3730777.7351
MOROCCO	1995	3469641.1344
MOROCCO	1994	3747593.2076
MOROCCO	1993	3620742.6983
MOROCCO	1992	4303609.2486
MOZAMBIQUE	1998	2024719.4607
MOZAMBIQUE	1997	3706003.0867
MOZAMBIQUE	1996	3376430.9303
MOZAMBIQUE	1995	2737631.6427
MOZAMBIQUE	1994	3373146.4811
MOZAMBIQUE	1993	3608300.3738
MOZAMBIQUE	1992	3551263.9502
PERU	1998	2142791.9724
PERU	1997	4664076.1540
PERU	1996	3623628.9338
PERU	1995	3908939.7912
PERU	1994	3386204.1565
PERU	1993	3877048.4889
PERU	1992	3768394.2488
ROMANIA	1998	1760625.7030
ROMANIA	1997	2707685.3292
ROMANIA	1996	2553345.4786
ROMANIA	1995	2715901.5896
ROMANIA	1994	3023644.0564
ROMANIA	1993	2873247.3205

```
ROMANIA              1992    2728060.7073
RUSSIA               1998    2975973.2167
RUSSIA               1997    3785806.4681
RUSSIA               1996    4217625.5866
RUSSIA               1995    3883445.5153
RUSSIA               1994    4395855.0063
RUSSIA               1993    3900944.1769
RUSSIA               1992    4691358.6091
SAUDI ARABIA         1998    2931482.8334
SAUDI ARABIA         1997    5498943.1556
SAUDI ARABIA         1996    4473723.7384
SAUDI ARABIA         1995    5939212.9339
SAUDI ARABIA         1994    4527695.7092
SAUDI ARABIA         1993    4928702.0169
SAUDI ARABIA         1992    5527261.5215
UNITED KINGDOM       1998    3198731.3729
UNITED KINGDOM       1997    4363882.7444
UNITED KINGDOM       1996    4730956.6742
UNITED KINGDOM       1995    4842014.5464
UNITED KINGDOM       1994    4912706.5567
UNITED KINGDOM       1993    4415255.9632
UNITED KINGDOM       1992    4375524.2303
UNITED STATES        1998    1892045.1604
UNITED STATES        1997    3102027.8595
UNITED STATES        1996    3334320.2579
UNITED STATES        1995    3168244.6043
UNITED STATES        1994    3296960.1009
UNITED STATES        1993    3558109.0546
UNITED STATES        1992    2755129.3878
VIETNAM              1998    2906627.0252
VIETNAM              1997    4544560.4478
VIETNAM              1996    4314258.9990
VIETNAM              1995    4365340.8614
VIETNAM              1994    3686987.7125
VIETNAM              1993    3764237.1787
VIETNAM              1992    3420922.0038

175 row(s) retrieved.


commit work;
Data committed.

Query: 9 Date: 1997-07-22 Time: 11:45:08.937  336938.937



Database closed.
```

C-10: Query 10

```
-- QUERY 10 RETURNED ITEM REPORTING QUERY
select
FIRST 20
        c_custkey,
        c_name,
        sum(l_extendedprice * (1 - l_discount)) as revenue,
        c_acctbal,
        n_name,
        c_address,
        c_phone,
        c_comment
from
        customer, order, lineitem, nation
where
        c_custkey = o_custkey and
        l_orderkey = o_orderkey and
        o_orderdate >= date('1993-10-01') and
        o_orderdate < date('1993-10-01') + interval (3)
month to month and
        l_returnflag = 'R' and
        c_nationkey = n_nationkey
group by
c_custkey,c_name,c_acctbal,c_phone,n_name,c_address,c_co
mment
order by revenue desc;


c_custkey  9722
c_name     Customer#000009722
revenue    464618.2584
c_acctbal  474.04
n_name     CANADA
c_address  l Mwzn4NAk6j
c_phone    13-518-602-8070
c_comment  5L 50Oy
RSgBAzPxmOSi5wk6xxOR7kh2nnPlgy7LBng2hOw5B01
RmCM120L24Pkg7PS
        1zwC11BCnz4L6i15PkixP26l66

c_custkey  12800
c_name     Customer#000012800
revenue    444265.6422
c_acctbal  1900.84
n_name     PERU
c_address  57zjB3CQx4P4OB2R2MBi2mwhSllM4mn 4 nC6
c_phone    27-142-205-3552
c_comment  0hwglS77RB56Rx436lQ0N16CxhOPnmyhgwz
5z64wnj1kiC4jL350mM41y71hNxBllPj
        yA4hiN1wzjjM7SCxAN244mk2A

c_custkey  1025
c_name     Customer#000001025
revenue    442028.0224
c_acctbal  3363.46
n_name     INDIA
c_address  lkiSn154M5ROi
c_phone    18-588-456-4616
c_comment
0Bl45z233Rniw0OO064nPBgP16kimO0y74iLh73g1N4
m310 jQ yQzPA50iC 3MA75g
        2Bj162Nw4P

c_custkey  13028
c_name     Customer#000013028
revenue    441692.2402
c_acctbal  -452.66
n_name     UNITED KINGDOM
c_address  yP714ORSNgNN2LA3L5B
c_phone    33-253-660-2127
c_comment  xPkmnhL2BkhkNyww4khlxwwAymN
h11PSjBCNMi50LkyOhO6CC 5nzOQCALzliOk2R66
        w l05hRPO3iSP

c_custkey  3694
c_name     Customer#000003694
revenue    438180.0696
c_acctbal  2960.44
n_name     UNITED KINGDOM
c_address  2CCklmCBOCC
c_phone    33-421-331-3127
c_comment  MzLxQxLlLx3MPxlAwg1B5kg61zxkPnk
xiAm6PhMMAAQ2nzN3S6zzgP x70w0lhhPx4Q
        RzlMMy0204lAl3mBO7jh2jAP0N60wg367z

c_custkey  976
c_name     Customer#000000976
revenue    435897.6317
c_acctbal  7772.85
```

n_name ROMANIA
c_address QzR 56Px1kgS wANnAz02RS 30n Pm
c_phone 29-436-660-4732
c_comment kzn32776
gwzkMzzzO4yxOAnkR7hR4R4x2SMwilz3x6h
nN7OnNLRMml3 kz5SLwi1yk
lOxiwS4g0wmA5A 4hmgBSwRRiQ1

c_custkey 8206
c_name Customer#000008206
revenue 429905.1096
c_acctbal 6046.36
n_name ARGENTINA
c_address P yMg30BBBBx NMgC03AmzN2
c_phone 11-571-859-1370
c_comment
hLi122RMPmLC36Oy0kxO71zz2wCR0QQCl7z26hlQ3mM

c_custkey 13532
c_name Customer#000013532
revenue 427731.8043
c_acctbal -924.18
n_name KENYA
c_address
6ij7M5PBMx2kwwyz62Oj4SL5S0mRCwl3m1Rmw
c_phone 24-525-332-7244
c_comment 7ih7yRz214zO67AiNPx64nO5l5k
yj6i3jLA5PCL15Q4QlA3ll60iM1P iBxCixg6 1h
Ch2RCnjOzk5R OnO 1OhhC3m4631m5

c_custkey 12745
c_name Customer#000012745
revenue 422327.6927
c_acctbal 9691.33
n_name CHINA
c_address SgS1LMC4gB2NM3wh
c_phone 28-985-189-6174
c_comment jl72wjSw0 S6 7L4Cgxw PkyO5Nl2LL7LBR

c_custkey 2344
c_name Customer#000002344
revenue 411240.1086
c_acctbal 5597.22
n_name MOROCCO
c_address O3PC7ikBgw OAzPAlm2P 426zm3BnBN6Q1O
6N
c_phone 25-593-745-7663
c_comment 5NBn0wRNngLw2z5kyn1AhL0ASyg6SMhM
i2kMOyxARAnlO0Q5j4CBNARix7ABlMAC

c_custkey 2656
c_name Customer#000002656
revenue 401185.9523
c_acctbal 8115.55
n_name ALGERIA
c_address On55lAS3Rm5RxS m
c_phone 10-667-469-8092
c_comment 46ABx4jgni mlBMPCLxRhyPQM4RNS
5yO1L7zSOmk MhPxAxQQ6lQnLj 17LymOhi4l5
innzOyB2Olxzmw3gmx0SxiyBN5CSMNgCkLCkMgO

c_custkey 59
c_name Customer#000000059
revenue 400759.1501
c_acctbal 3458.60
n_name ARGENTINA
c_address wP6CMyClly0lS4CAM1mzm
c_phone 11-355-584-3112
c_comment l1g7xBCxxC7SM
5AkmnmAk00677O1MzA2R7A0Cx0Njixj56jL2iN
PNkSNQiy55m6ki3
OgnhM47mSR7B

c_custkey 7069
c_name Customer#000007069
revenue 396217.5195
c_acctbal 8198.94
n_name INDONESIA
c_address 55Cw7ChL4Bi5ONn2A4m2i2n4nSNQQMjml
c_phone 19-644-744-1798
c_comment
6jNS624l75zlxNli4lxO5zyPykPS1xnilS0NhkgOAkSx7P

c_custkey 6553
c_name Customer#000006553
revenue 385863.5946
c_acctbal 8985.90
n_name MOZAMBIQUE
c_address R3LnnxONBjCLC0MRkxy7
c_phone 26-166-724-4677
c_comment S7CkNLwA3kh006j71lwAlC25Bw6AMQ6i
6C0OSS6O7ARNNny60Ogh 3642mRxyiAgy5
yk 3nPO4473wkNg5R6gzO4lz3zmM2m7MiLAilCC

c_custkey 3095
c_name Customer#000003095
revenue 384246.1083
c_acctbal 8829.21
n_name IRAQ
c_address S1gMCnBLwzi mCgB664 jl00Ll1Snh1iPMgCgR5
c_phone 21-847-218-8188
c_comment 3LSx7PxS
A4A5Cl3gAy3mg4Qj2xQlyx7xM1kA664AM7zmMmzORh3
C1h MO3nw6Mymilj
AMg65hOMB4Sn44kO w0lin7

c_custkey 3391
c_name Customer#000003391
revenue 382541.7762
c_acctbal 7742.35
n_name CANADA
c_address m3 CORmQNLzkShymLS iMkCimRSl20 NB
c_phone 13-592-494-2668
c_comment ynMlmhMBA5ikC1nCghlmAhQ0
675S3y2R33yjkNPQOS

c_custkey 13678
c_name Customer#000013678
revenue 376280.5564
c_acctbal 9030.40
n_name MOROCCO
c_address BMk77lQm1lwNA0LghAkg3hCwNl4
c_phone 25-306-951-3937
c_comment
mOS55RASx1wP136nQ5xBLznLhgw1kQ6PO6imNxQ7kR0
x71P0SzByMzh

c_custkey 6062
c_name Customer#000006062
revenue 374512.6544
c_acctbal 1370.35
n_name CANADA
c_address n5zzil60zyxAlkzx7x1nihigPzR OBkR znMOMh
c_phone 13-756-700-4918
c_comment 4zAm4wNB
li4QRPgPz2wM541x043hmLj4O3LBkALCP16hj2RQBO1O
MNly7ww1QP7w5i
SSn0jNhAR yQmmz1hi5j3

c_custkey 554
c_name Customer#000000554
revenue 373004.4702

```
c_acctbal  8395.57
n_name     BRAZIL
c_address  jC5zhQky4zQB27lB5Sm AQhQ  Px0
c_phone    12-938-503-7317
c_comment  0nxCl3 xSmiLQO 1M
2n0NCiRlnMMxP25j26x2igLhNOxjgMgmwwy7OkjzCACOg
0z2LA
         jOm0RPRmOPiCAAQwLlQSg 1yS3
gLCM1M2BzjnSjPl3nwAkk

c_custkey  13126
c_name     Customer#000013126
revenue    371722.0011
c_acctbal  6172.91
n_name     INDIA
c_address   xPAS4MnPh40i5Q2h4NQ61zz4RkyAwANA
c_phone    18-288-190-4145
c_comment  nniMkAN6C0ClQ0mMmPz27liz4hk6L
2MlwPxh42N1l0R2hRwxxzlwMkxO4MAyz7RCj43
         NxLwQ3m6P27yAj

20 row(s) retrieved.


commit work;
Data committed.

Query: 10 Date: 1997-07-22 Time: 11:44:20.890
336890.890


begin work;
Started transaction.


-- using default substitutions
```

C-11: Query 11

```
-- QUERY 11 IMPORTANT STOCK IDENTIFICATION
QUERY

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) *
0.0010000000
            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

     12098     16227681.21
      5134     15709338.52
     13334     15023662.41
     17052     14351644.20
      3452     14070870.14
     12552     13332469.18
      1084     13170428.29
      5797     13038622.72
     12633     12892561.61
       403     12856217.34
      1833     12024581.72
      2084     11502875.36
     17349     11354213.05
     18427     11282385.24
      2860     11262529.95
     17852     10934711.93
      9871     10889253.68
     12231     10841131.39
      6366     10759786.81
     12146     10257362.66
      5043     10226395.88
     12969     10125777.93

22 row(s) retrieved.


commit work;
Data committed.

Query: 11 Date: 1997-07-22 Time: 11:44:22.687
336892.687


begin work;
Started transaction.


-- using default substitutions
```

C-12: Query 12

```
-- QUERY 12 SHIPPING MODES AND ORDER PRIORITY
QUERY
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
            order, 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 >= date('1994-01-01') and
                        l_receiptdate < date('1994-01-01') +
interval (1) year to year
group by l_shipmode
order by l_shipmode;

l_shipmode  high_line_count   low_line_count

MAIL                 654            950
SHIP                 684           1004

2 row(s) retrieved.
```

```
commit work;
Data committed.

Query: 12 Date: 1997-07-22 Time: 11:44:43.218
336913.218


begin work;
Started transaction.


-- using default substitutions
```

C-13: Query 13

```
-- QUERY 13 SALES CLERK PERFORMANCE QUERY
select
        year(o_orderdate) as year,
        sum(l_extendedprice * (1 - l_discount)) as revenue
from
        lineitem, order
where
        o_orderkey = l_orderkey and
        o_clerk = 'Clerk#000000088' and
        l_returnflag = 'R'
group by 1
order by 1;

 year        revenue

 1992    1262855.7306
 1993     964121.0328
 1994    1750395.2936
 1995     198820.2992

4 row(s) retrieved.


commit work;
Data committed.

Query: 13 Date: 1997-07-22 Time: 11:44:56.828
336926.828


begin work;
Started transaction.


-- using default substitutions
```

C-14: Query 14

```
-- QUERY 14 PROMOTION EFFECT QUERY
select
    100.0 * 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 to month;

  promo_revenue

        16.73

1 row(s) retrieved.


commit work;
Data committed.

Query: 14 Date: 1997-07-22 Time: 11:44:31.437
336901.437


begin work;
Started transaction.


-- using default substitutions
```

C-15: Query 15b

```
-- QUERY 15b  TOP SUPPLIER QUERY
create table revenue0
(supplier_no integer,
total_revenue decimal(13,3))
fragment by round robin in o_okey1;
Table created.



insert into revenue0
select
        l_suppkey,
        sum(l_extendedprice * (1 - l_discount))
from lineitem
where
        l_shipdate >= date('1996-01-01') and
        l_shipdate < date('1996-01-01') + interval (3)
month to month
group by l_suppkey;
1000 row(s) inserted.



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


s_suppkey     389
s_name        Supplier#000000389
s_address     PB1Lx0xx6LMz3h7Rx63m6j3QmMx
s_phone       34-885-883-5717
total_revenue  1418538.214

1 row(s) retrieved.
```

```
drop table revenue0;
Table dropped.


commit work;
Data committed.

Query: 15b Date: 1997-07-22 Time: 11:44:18.828
336888.828


begin work;
Started transaction.


-- using default substitutions
```

C-16: Query 16

```
-- QUERY 16 PARTS/SUPPLIER RELATIONSHIP QUERY
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 '%Better Business
Bureau%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#14   SMALL ANODIZED NICKEL            45
12
Brand#22   SMALL BURNISHED BRASS            19
12
Brand#25   PROMO POLISHED COPPER            14
12
Brand#35   LARGE ANODIZED STEEL             45
12
Brand#35   PROMO BRUSHED COPPER              9
12
Brand#51   ECONOMY ANODIZED STEEL            9
12
Brand#53   LARGE BRUSHED NICKEL             45
12
Brand#11   ECONOMY POLISHED COPPER           14
8
Brand#11   LARGE PLATED STEEL              23          8
Brand#11   PROMO POLISHED STEEL             23
8
Brand#11   STANDARD ANODIZED COPPER          9
8
Brand#12   ECONOMY BURNISHED BRASS           9
8
Brand#12   LARGE ANODIZED BRASS             14
8
Brand#12   SMALL ANODIZED TIN               23          8
Brand#12   SMALL BRUSHED NICKEL             23
8
Brand#12   STANDARD ANODIZED BRASS           3
8
Brand#12   STANDARD BURNISHED TIN           23
8
Brand#13   ECONOMY POLISHED BRASS           9
8
Brand#13   LARGE BURNISHED COPPER           45
8
Brand#13   MEDIUM ANODIZED STEEL            23
8
Brand#13   MEDIUM PLATED NICKEL             3
8
Brand#13   PROMO BURNISHED BRASS            9
8
Brand#13   PROMO POLISHED BRASS             3
8
Brand#13   PROMO POLISHED TIN               36          8
Brand#13   SMALL BURNISHED STEEL            23
8
Brand#13   STANDARD BRUSHED STEEL           9
8
Brand#14   ECONOMY BRUSHED TIN              3
8
Brand#14   ECONOMY BURNISHED TIN            23
8
Brand#14   PROMO BRUSHED STEEL              9
8
Brand#14   PROMO PLATED TIN                 45          8
Brand#15   ECONOMY PLATED TIN               9
8
Brand#15   STANDARD BRUSHED COPPER          14
8
Brand#15   STANDARD PLATED TIN              3
8
Brand#21   ECONOMY POLISHED TIN             3
8
Brand#21   PROMO POLISHED COPPER            9
8
Brand#21   PROMO POLISHED TIN               49          8
Brand#21   STANDARD PLATED BRASS            49
8
Brand#21   STANDARD PLATED NICKEL           49
8
Brand#22   ECONOMY ANODIZED TIN             49
8
Brand#22   ECONOMY BRUSHED BRASS            14
8
Brand#22   LARGE BURNISHED TIN              36
8
Brand#22   MEDIUM ANODIZED STEEL            36
8
Brand#22   MEDIUM PLATED STEEL              9
8
Brand#22   PROMO POLISHED NICKEL            9
8
Brand#22   SMALL ANODIZED STEEL             19
8
Brand#22   STANDARD ANODIZED COPPER          23
8
Brand#23   ECONOMY BRUSHED NICKEL           23
8
Brand#23   LARGE ANODIZED BRASS             9
8
Brand#23   LARGE ANODIZED STEEL             23
8
Brand#23   SMALL BRUSHED COPPER             23
8
Brand#23   STANDARD BRUSHED TIN             3
```

8
Brand#23 STANDARD BURNISHED NICKEL 49
8
Brand#23 STANDARD PLATED NICKEL 36
8
Brand#24 ECONOMY ANODIZED BRASS 19
8
Brand#24 ECONOMY POLISHED BRASS 36
8
Brand#24 LARGE BURNISHED STEEL 14
8
Brand#24 MEDIUM PLATED NICKEL 36
8
Brand#25 ECONOMY BRUSHED STEEL 49
8
Brand#25 MEDIUM BURNISHED TIN 3
8
Brand#25 PROMO ANODIZED TIN 36
8
Brand#25 PROMO PLATED NICKEL 3
8
Brand#25 SMALL BURNISHED BRASS 3
8
Brand#31 LARGE ANODIZED BRASS 3
8
Brand#31 SMALL ANODIZED COPPER 3
8
Brand#31 SMALL ANODIZED NICKEL 9
8
Brand#31 SMALL ANODIZED STEEL 14
8
Brand#32 MEDIUM ANODIZED STEEL 49
8
Brand#32 MEDIUM BURNISHED COPPER 19
8
Brand#32 SMALL BURNISHED STEEL 23
8
Brand#32 STANDARD BURNISHED STEEL 45
8
Brand#34 ECONOMY ANODIZED NICKEL 49
8
Brand#34 LARGE BURNISHED TIN 49
8
Brand#34 PROMO ANODIZED TIN 3 8
Brand#34 SMALL BRUSHED TIN 3 8
Brand#34 STANDARD BURNISHED TIN 23
8
Brand#35 MEDIUM BRUSHED STEEL 45
8
Brand#35 PROMO BURNISHED STEEL 14
8
Brand#35 SMALL BURNISHED STEEL 23
8
Brand#35 SMALL POLISHED COPPER 14
8
Brand#35 STANDARD PLATED COPPER 9
8
Brand#41 ECONOMY BRUSHED BRASS 23
8
Brand#41 LARGE BURNISHED STEEL 23
8
Brand#41 PROMO BURNISHED TIN 14
8
Brand#41 PROMO PLATED STEEL 36
8
Brand#41 PROMO POLISHED TIN 19 8
Brand#41 SMALL BURNISHED COPPER 23
8
Brand#42 LARGE POLISHED TIN 14 8
Brand#42 MEDIUM ANODIZED TIN 49
8
Brand#42 MEDIUM BRUSHED TIN 14
8
Brand#42 MEDIUM BURNISHED NICKEL 23
8
Brand#42 MEDIUM PLATED COPPER 45
8
Brand#42 MEDIUM PLATED TIN 45 8
Brand#42 SMALL PLATED COPPER 36
8
Brand#43 ECONOMY BRUSHED STEEL 45
8
Brand#43 LARGE BRUSHED COPPER 19
8
Brand#43 PROMO BRUSHED BRASS 36
8
Brand#43 SMALL BURNISHED TIN 45
8
Brand#43 SMALL PLATED COPPER 45
8
Brand#44 PROMO POLISHED TIN 23 8
Brand#44 SMALL POLISHED NICKEL 14
8
Brand#44 SMALL POLISHED TIN 45 8
Brand#44 STANDARD BURNISHED COPPER 3
8
Brand#51 LARGE ANODIZED BRASS 19
8
Brand#51 LARGE POLISHED COPPER 23
8
Brand#51 MEDIUM ANODIZED TIN 9
8
Brand#51 MEDIUM ANODIZED TIN 14
8
Brand#51 MEDIUM BURNISHED NICKEL 23
8
Brand#51 SMALL ANODIZED COPPER 45
8
Brand#51 SMALL ANODIZED COPPER 49
8
Brand#51 SMALL BRUSHED COPPER 45
8
Brand#51 SMALL BRUSHED TIN 36 8
Brand#51 STANDARD POLISHED TIN 3
8
Brand#52 ECONOMY ANODIZED STEEL 3
8
Brand#52 ECONOMY PLATED TIN 19
8
Brand#52 LARGE PLATED TIN 3 8
Brand#52 MEDIUM ANODIZED TIN 19
8
Brand#52 MEDIUM BURNISHED COPPER 3
8
Brand#52 PROMO POLISHED BRASS 23
8
Brand#52 SMALL PLATED COPPER 36
8
Brand#52 SMALL POLISHED NICKEL 9
8
Brand#52 STANDARD POLISHED NICKEL 45
8
Brand#53 ECONOMY POLISHED STEEL 45
8
Brand#53 LARGE POLISHED NICKEL 3
8
Brand#53 SMALL BRUSHED COPPER 14
8
Brand#53 STANDARD PLATED STEEL 45
8
Brand#54 ECONOMY POLISHED BRASS 49
8

Brand#54 ECONOMY POLISHED TIN 23
8
Brand#54 MEDIUM BRUSHED STEEL 9
8
Brand#54 SMALL BURNISHED NICKEL 14
8
Brand#54 SMALL PLATED TIN 14 8
Brand#54 STANDARD BURNISHED STEEL 14
8
Brand#54 STANDARD PLATED BRASS 23
8
Brand#55 MEDIUM BURNISHED TIN 36
8
Brand#55 PROMO ANODIZED BRASS 14
8
Brand#55 STANDARD BURNISHED COPPER 45
8
Brand#15 STANDARD PLATED TIN 36
7
Brand#21 SMALL POLISHED STEEL 3
7
Brand#23 SMALL POLISHED BRASS 49
7
Brand#34 MEDIUM BURNISHED NICKEL 3
7
Brand#42 STANDARD PLATED COPPER 19
7
Brand#51 LARGE POLISHED NICKEL 14
7
Brand#54 LARGE ANODIZED NICKEL 49
7
Brand#11 ECONOMY ANODIZED BRASS 19
4
Brand#11 ECONOMY ANODIZED BRASS 45
4
Brand#11 ECONOMY ANODIZED NICKEL 36
4
Brand#11 ECONOMY BRUSHED COPPER 3
4
Brand#11 ECONOMY BRUSHED COPPER 9
4
Brand#11 ECONOMY BRUSHED STEEL 9
4
Brand#11 ECONOMY BRUSHED STEEL 36
4
Brand#11 ECONOMY BURNISHED BRASS 36
4
Brand#11 ECONOMY BURNISHED COPPER 9
4
Brand#11 ECONOMY BURNISHED COPPER 49
4
Brand#11 ECONOMY BURNISHED NICKEL 14
4
Brand#11 ECONOMY BURNISHED NICKEL 49
4
Brand#11 ECONOMY PLATED COPPER 19
4
Brand#11 ECONOMY PLATED NICKEL 45
4
Brand#11 ECONOMY PLATED TIN 9
4
Brand#11 ECONOMY POLISHED BRASS 3
4
Brand#11 ECONOMY POLISHED COPPER 3
4
Brand#11 ECONOMY POLISHED COPPER 45
4
Brand#11 ECONOMY POLISHED NICKEL 36
4
Brand#11 ECONOMY POLISHED STEEL 23
4
Brand#11 ECONOMY POLISHED TIN 14
4
Brand#11 LARGE ANODIZED COPPER 23
4
Brand#11 LARGE ANODIZED NICKEL 9
4
Brand#11 LARGE ANODIZED STEEL 9
4
Brand#11 LARGE BRUSHED STEEL 19
4
Brand#11 LARGE BRUSHED TIN 3 4
Brand#11 LARGE BRUSHED TIN 14 4
Brand#11 LARGE BURNISHED COPPER 9
4
Brand#11 LARGE BURNISHED COPPER 19
4
Brand#11 LARGE BURNISHED STEEL 23
4
Brand#11 LARGE BURNISHED TIN 9 4
Brand#11 LARGE PLATED COPPER 23
4
Brand#11 LARGE PLATED TIN 9 4
Brand#11 LARGE PLATED TIN 14 4
Brand#11 LARGE PLATED TIN 23 4
Brand#11 LARGE POLISHED NICKEL 49
4
Brand#11 MEDIUM ANODIZED BRASS 45
4
Brand#11 MEDIUM ANODIZED TIN 14
4
Brand#11 MEDIUM BRUSHED BRASS 14
4
Brand#11 MEDIUM BRUSHED BRASS 45
4
Brand#11 MEDIUM BRUSHED NICKEL 14
4
Brand#11 MEDIUM BRUSHED NICKEL 36
4
Brand#11 MEDIUM BRUSHED STEEL 19
4
Brand#11 MEDIUM BURNISHED COPPER 9
4
Brand#11 MEDIUM BURNISHED TIN 36
4
Brand#11 MEDIUM PLATED BRASS 3
4
Brand#11 MEDIUM PLATED TIN 19 4
Brand#11 PROMO ANODIZED BRASS 3
4
Brand#11 PROMO ANODIZED BRASS 19
4
Brand#11 PROMO ANODIZED BRASS 45
4
Brand#11 PROMO ANODIZED BRASS 49
4
Brand#11 PROMO ANODIZED STEEL 23
4
Brand#11 PROMO ANODIZED TIN 45
4
Brand#11 PROMO BRUSHED BRASS 23
4
Brand#11 PROMO BRUSHED STEEL 3
4
Brand#11 PROMO BURNISHED BRASS 23
4
Brand#11 PROMO BURNISHED BRASS 36
4
Brand#11 PROMO BURNISHED BRASS 49
4
Brand#11 PROMO BURNISHED TIN 9
4

Brand#11 PROMO PLATED BRASS 9 4
Brand#11 PROMO PLATED BRASS 45
4
Brand#11 PROMO PLATED NICKEL 19
4
Brand#11 PROMO POLISHED BRASS 3
4
Brand#11 PROMO POLISHED BRASS 9
4
Brand#11 PROMO POLISHED BRASS 19
4
Brand#11 PROMO POLISHED COPPER 14
4
Brand#11 PROMO POLISHED COPPER 45
4
Brand#11 PROMO POLISHED TIN 49 4
Brand#11 SMALL ANODIZED COPPER 36
4
Brand#11 SMALL ANODIZED NICKEL 3
4
Brand#11 SMALL ANODIZED NICKEL 14
4
Brand#11 SMALL ANODIZED TIN 14 4
Brand#11 SMALL ANODIZED TIN 19 4
Brand#11 SMALL ANODIZED TIN 45 4
Brand#11 SMALL BRUSHED TIN 14 4
Brand#11 SMALL BRUSHED TIN 23 4
Brand#11 SMALL BRUSHED TIN 45 4
Brand#11 SMALL BURNISHED BRASS 49
4
Brand#11 SMALL BURNISHED COPPER 23
4
Brand#11 SMALL PLATED COPPER 45
4
Brand#11 SMALL PLATED NICKEL 3 4
Brand#11 SMALL PLATED STEEL 36 4
Brand#11 SMALL PLATED TIN 19 4
Brand#11 SMALL POLISHED BRASS 14
4
Brand#11 SMALL POLISHED BRASS 23
4
Brand#11 SMALL POLISHED COPPER 14
4
Brand#11 SMALL POLISHED COPPER 36
4
Brand#11 SMALL POLISHED STEEL 9
4
Brand#11 STANDARD BRUSHED COPPER 23
4
Brand#11 STANDARD BRUSHED NICKEL 14
4
Brand#11 STANDARD BRUSHED TIN 14
4
Brand#11 STANDARD BURNISHED BRASS 3
4
Brand#11 STANDARD BURNISHED STEEL 23
4
Brand#11 STANDARD PLATED BRASS 19
4
Brand#11 STANDARD PLATED TIN 19
4
Brand#11 STANDARD POLISHED NICKEL 45
4
Brand#11 STANDARD POLISHED TIN 14
4
Brand#11 STANDARD POLISHED TIN 45
4
Brand#12 ECONOMY ANODIZED BRASS 23
4
Brand#12 ECONOMY ANODIZED COPPER 14
4
Brand#12 ECONOMY ANODIZED NICKEL 19
4
Brand#12 ECONOMY ANODIZED NICKEL 45
4
Brand#12 ECONOMY ANODIZED STEEL 9
4
Brand#12 ECONOMY BRUSHED COPPER 36
4
Brand#12 ECONOMY BRUSHED NICKEL 49
4
Brand#12 ECONOMY BRUSHED STEEL 49
4
Brand#12 ECONOMY BURNISHED COPPER 45
4
Brand#12 ECONOMY PLATED COPPER 23
4
Brand#12 ECONOMY PLATED STEEL 23
4
Brand#12 ECONOMY PLATED TIN 36
4
Brand#12 ECONOMY POLISHED BRASS 14
4
Brand#12 ECONOMY POLISHED COPPER 45
4
Brand#12 ECONOMY POLISHED NICKEL 9
4
Brand#12 LARGE ANODIZED NICKEL 9
4
Brand#12 LARGE ANODIZED NICKEL 49
4
Brand#12 LARGE ANODIZED STEEL 49
4
Brand#12 LARGE ANODIZED TIN 36 4
Brand#12 LARGE ANODIZED TIN 45 4
Brand#12 LARGE BURNISHED BRASS 14
4
Brand#12 LARGE BURNISHED BRASS 19
4
Brand#12 LARGE BURNISHED COPPER 9
4
Brand#12 LARGE BURNISHED NICKEL 45
4
Brand#12 LARGE BURNISHED TIN 36
4
Brand#12 LARGE PLATED BRASS 3 4
Brand#12 LARGE PLATED STEEL 36 4
Brand#12 LARGE PLATED STEEL 45 4
Brand#12 LARGE PLATED TIN 23 4
Brand#12 LARGE POLISHED COPPER 14
4
Brand#12 LARGE POLISHED COPPER 19
4
Brand#12 LARGE POLISHED COPPER 49
4
Brand#12 LARGE POLISHED STEEL 3
4
Brand#12 MEDIUM ANODIZED COPPER 9
4
Brand#12 MEDIUM ANODIZED COPPER 45
4
Brand#12 MEDIUM ANODIZED NICKEL 45
4
Brand#12 MEDIUM BRUSHED BRASS 19
4
Brand#12 MEDIUM BRUSHED COPPER 9
4
Brand#12 MEDIUM BRUSHED COPPER 36
4
Brand#12 MEDIUM BRUSHED COPPER 49
4
Brand#12 MEDIUM BRUSHED NICKEL 3

4
Brand#12 MEDIUM BRUSHED NICKEL 14
4
Brand#12 MEDIUM BRUSHED NICKEL 23
4
Brand#12 MEDIUM BURNISHED BRASS 3
4
Brand#12 MEDIUM BURNISHED COPPER 36
4
Brand#12 MEDIUM BURNISHED NICKEL 19
4
Brand#12 MEDIUM BURNISHED TIN 14
4
Brand#12 MEDIUM PLATED BRASS 23
4
Brand#12 MEDIUM PLATED TIN 19 4
Brand#12 MEDIUM PLATED TIN 23 4
Brand#12 PROMO ANODIZED BRASS 9
4
Brand#12 PROMO ANODIZED BRASS 45
4
Brand#12 PROMO ANODIZED NICKEL 14
4
Brand#12 PROMO ANODIZED STEEL 49
4
Brand#12 PROMO ANODIZED TIN 3 4
Brand#12 PROMO ANODIZED TIN 19
4
Brand#12 PROMO BRUSHED COPPER 14
4
Brand#12 PROMO BRUSHED COPPER 19
4
Brand#12 PROMO BRUSHED NICKEL 23
4
Brand#12 PROMO BRUSHED STEEL 23
4
Brand#12 PROMO BRUSHED STEEL 36
4
Brand#12 PROMO BURNISHED BRASS 49
4
Brand#12 PROMO BURNISHED TIN 9
4
Brand#12 PROMO BURNISHED TIN 14
4
Brand#12 PROMO PLATED BRASS 36
4
Brand#12 PROMO POLISHED COPPER 23
4
Brand#12 PROMO POLISHED NICKEL 3
4
Brand#12 PROMO POLISHED NICKEL 9
4
Brand#12 PROMO POLISHED STEEL 14
4
Brand#12 PROMO POLISHED TIN 23 4
Brand#12 PROMO POLISHED TIN 36 4
Brand#12 SMALL ANODIZED BRASS 36
4
Brand#12 SMALL ANODIZED COPPER 23
4
Brand#12 SMALL ANODIZED STEEL 36
4
Brand#12 SMALL ANODIZED TIN 14 4
Brand#12 SMALL BRUSHED COPPER 19
4
Brand#12 SMALL BRUSHED COPPER 36
4
Brand#12 SMALL BRUSHED TIN 36 4
Brand#12 SMALL BURNISHED BRASS 14
4
Brand#12 SMALL BURNISHED COPPER 9
4
Brand#12 SMALL BURNISHED COPPER 36
4
Brand#12 SMALL PLATED BRASS 9 4
Brand#12 SMALL POLISHED BRASS 49
4
Brand#12 SMALL POLISHED NICKEL 19
4
Brand#12 SMALL POLISHED TIN 3 4
Brand#12 STANDARD ANODIZED BRASS 19
4
Brand#12 STANDARD ANODIZED NICKEL 19
4
Brand#12 STANDARD ANODIZED STEEL 19
4
Brand#12 STANDARD BRUSHED COPPER 36
4
Brand#12 STANDARD BRUSHED NICKEL 23
4
Brand#12 STANDARD BRUSHED STEEL 49
4
Brand#12 STANDARD BURNISHED BRASS 23
4
Brand#12 STANDARD BURNISHED COPPER 14
4
Brand#12 STANDARD BURNISHED NICKEL 45
4
Brand#12 STANDARD BURNISHED NICKEL 49
4
Brand#12 STANDARD BURNISHED TIN 3
4
Brand#12 STANDARD BURNISHED TIN 14
4
Brand#12 STANDARD PLATED BRASS 19
4
Brand#12 STANDARD PLATED NICKEL 45
4
Brand#12 STANDARD PLATED STEEL 36
4
Brand#12 STANDARD PLATED STEEL 45
4
Brand#12 STANDARD PLATED TIN 9
4
Brand#12 STANDARD POLISHED BRASS 49
4
Brand#12 STANDARD POLISHED COPPER 3
4
Brand#12 STANDARD POLISHED NICKEL 23
4
Brand#12 STANDARD POLISHED TIN 14
4
Brand#13 ECONOMY ANODIZED NICKEL 14
4
Brand#13 ECONOMY ANODIZED NICKEL 19
4
Brand#13 ECONOMY ANODIZED STEEL 45
4
Brand#13 ECONOMY ANODIZED STEEL 49
4
Brand#13 ECONOMY BRUSHED BRASS 3
4
Brand#13 ECONOMY BURNISHED STEEL 14
4
Brand#13 ECONOMY BURNISHED TIN 19
4
Brand#13 ECONOMY BURNISHED TIN 45
4
Brand#13 ECONOMY PLATED COPPER 19
4
Brand#13 ECONOMY PLATED NICKEL 3
4

```
Brand#13   ECONOMY PLATED STEEL          23
4
Brand#13   ECONOMY PLATED TIN            3
4
Brand#13   ECONOMY POLISHED BRASS        3
4
Brand#13   ECONOMY POLISHED COPPER       9
4
Brand#13   ECONOMY POLISHED COPPER       49
4
Brand#13   ECONOMY POLISHED STEEL        23
4
Brand#13   ECONOMY POLISHED STEEL        49
4
Brand#13   LARGE ANODIZED BRASS          23
4
Brand#13   LARGE ANODIZED COPPER         19
4
Brand#13   LARGE ANODIZED NICKEL         9
4
Brand#13   LARGE ANODIZED STEEL          45
4
Brand#13   LARGE ANODIZED TIN            19          4
Brand#13   LARGE BRUSHED BRASS           3
4
Brand#13   LARGE BRUSHED BRASS           9
4
Brand#13   LARGE BRUSHED BRASS           19
4
Brand#13   LARGE BRUSHED COPPER          9
4
Brand#13   LARGE BRUSHED COPPER          36
4
Brand#13   LARGE BRUSHED NICKEL          3
4
Brand#13   LARGE BRUSHED NICKEL          9
4
Brand#13   LARGE BRUSHED NICKEL          14
4
```

C-17: Query 17a

```
-- QUERY 17a MINIMUM COST SUPPLIER QUERY
create table avg_quantity0
(partkey integer,
avgqty decimal(13,3)
) fragment by round robin in o_okey1;
Table created.



insert into avg_quantity0
select
        p_partkey,
        0.2 * avg(l_quantity)
from
        part, lineitem
where
        p_partkey=l_partkey and
        p_brand= 'Brand#23' and
        p_container= 'MED BOX'
group by p_partkey;
18 row(s) inserted.



select
        sum(l_extendedprice)/7.0 as avg_yearly
from
        lineitem, avg_quantity0
where
        partkey = l_partkey and
        l_quantity < avgqty;

  avg_yearly

   24436.88

1 row(s) retrieved.



drop table avg_quantity0;
Table dropped.


commit work;
Data committed.

Query: 17a Date: 1997-07-22 Time: 11:44:48.62  336918.062


begin work;
Started transaction.


-- using default substitutions
```

Appendix D: Substitution Parameters and Seeds

D-1: Query Substitution Parameters

1	119				
4	1997-10-01				
15	1997-10-01				
10	1995-01-01				
11	UNITED STATES	0.0000033333			
6	1997-01-01	0.07	24		
2	50	COPPER	ASIA		
16	Brand#45	MEDIUM PLATED	42	41	
	50	30	37	5	24
	11				
14	1997-12-01				
8	UNITED STATES	AMERICA			
	LARGE POLISHED COPPER				
12	SHIP	MAIL	1995-01-01		
17	Brand#45	MED CASE			
3	MACHINERY	1995-03-20			
5	MIDDLE EAST	1996-01-01			
13	Clerk#000000981				
7	UNITED STATES	MOROCCO			
9	white				

D-2: RNG Seed

Note: The following is an exerpt from the EQT submitted to DBACCESS. It includes the DBGEN-generated comment with the seed used for the performance runs.

```
database dssf100;
set pdqpriority 100;
set isolation repeatable read;
begin work;
-- using 160797 as a seed to the RNG
```

Appendix E: Implementation Specific Layer and Drivers

E-1: driver.sh

```
#!sh

if [ $# != 4 ]
then
        echo "Usage:  $0 run_id seed pv scale"
        echo "        Use DDMMYYYY for seed"
        exit
fi

TOP="/tpcd"

runid=$1
INFORMIXDIR="e:/informix"
SQEXPLN="${INFORMIXDIR}/sqexpln/informix.out"
DBGENDIR="dbgen"
AUDIT_TAG=""
DSS_QUERY="$TOP/queries2"
DSS_CONFIG="$TOP/${DBGENDIR}"
QUERY_DIR="$TOP/queries_mj"
RES_DIR="$TOP/results/res_${1}"
PATH="$TOP/bin;$TOP;$PATH;$INFORMIXDIR/bin" ;
export PATH
QGEN="$TOP/${DBGENDIR}/qgen"
QUERIES="1 4 15b 10 11 6 2 16 14 8 12 17a 3 5 13 7 9"
SEED=$2
SCALE=$4
export DSS_QUERY DSS_CONFIG
if [ -f $SQEXPLN ]
then
  rm $SQEXPLN
fi

if [ ! -d $RES_DIR ]
then
        mkdir $RES_DIR
else rm $RES_DIR/*
fi
# echo "Perf/Validate? [P/V]: "
# read type
type=$3
if [ $type = "p" -o $type = "P" ]
then
type=P
dbname=dssf$SCALE
else
type=V
dbname=dssfq
SCALE=0.1
flags="-d"
fi

echo "database $dbname;" > ${RES_DIR}/session_input
echo "set pdqpriority 100;" >> ${RES_DIR}/session_input
echo "set isolation repeatable read;" >>
${RES_DIR}/session_input

# generate new EQT
for q in $QUERIES
do
   cat ${DSS_QUERY}/Start_query >>
${RES_DIR}/session_input
   $QGEN -c -s ${SCALE} -r $SEED ${flags} \
      -l ${RES_DIR}/params $q > ${QUERY_DIR}/${q}.sql
   cat ${QUERY_DIR}/${q}.sql >>
${RES_DIR}/session_input
   cat ${DSS_QUERY}/End_query >>
${RES_DIR}/session_input
echo "!timer $q" >> ${RES_DIR}/session_input

done

# execut performance tests runs
for run in 1
do
   echo "-- SF $SCALE"  >> ${RES_DIR}/Timing
   timer Stream1 >> ${RES_DIR}/Timing

   if [ $type != "V" ]
   then uf1.bat $SCALE >  ${RES_DIR}/uf1.out 2>&1
   fi
   sleep 5
   timer UF1 >> ${RES_DIR}/Timing
if [ -f $SQEXPLN ]
then
  cp $SQEXPLN ${RES_DIR}/uf1.plan
  rm $SQEXPLN
fi

   cat ${RES_DIR}/session_input | e:/informix/bin/dbaccess -e
- - > ${RES_DIR}/session_out 2>&1
if [ -f $SQEXPLN ]
then
  cp $SQEXPLN ${RES_DIR}/queries.plan
  rm $SQEXPLN
fi
   if [ $type != "V" ]
        then uf2.bat $SCALE >  ${RES_DIR}/uf2.out
2>&1
        fi
   sleep 5
   timer UF2 >> $RES_DIR/Timing
if [ -f $SQEXPLN ]
then
  cp $SQEXPLN ${RES_DIR}/uf2.plan
  rm $SQEXPLN
fi


# Calculate TPCD Numbers
head -1 $RES_DIR/Timing > $RES_DIR/tpcdcalc.in
grep Stream1 $RES_DIR/Timing >> $RES_DIR/tpcdcalc.in
grep UF1 $RES_DIR/Timing >> $RES_DIR/tpcdcalc.in
grep Time: $RES_DIR/session_out >> $RES_DIR/tpcdcalc.in
grep UF2 $RES_DIR/Timing >> $RES_DIR/tpcdcalc.in
tpcdcalc <  ${RES_DIR}/tpcdcalc.in >  ${RES_DIR}/result
cat ${RES_DIR}/result



cat ${RES_DIR}/session_out | awk -f postproc.awk
DIR=$RES_DIR
        if [ $type != "V" ]
        then
   uf1_reset.bat $SCALE > ${RES_DIR}/uf1_reset.out 2>&1
   uf2_reset.bat $SCALE > ${RES_DIR}/uf2_reset.out 2>&1
        fi

   for q in $QUERIES
   do
        head -400 ${RES_DIR}/${q}.out >
${RES_DIR}/m${runid}pqry${q}
   done
done
```

E-2: calc.c

```
#include <stdio.h>
#include <math.h>
#define N 19
#define QUERIES 17
main()
{
   int i;
   float time, sum, SF, UF1, UF2, qppd, qthd, qts;
   char qid[128];

   sum = 0.0;
   qts = 0.0;
   scanf ("SF %f\n", &SF);
   scanf ("UF1 %f\n", &UF1);
   printf ("\nScale Factor = %.2f\n", SF);
   printf ("Query\t Time (secs)\n");
   printf ("--------------------\n");
   printf ("UF1\t %8.2f\n", UF1);
   qts += UF1 ;
   sum += log(UF1);
   for (i = 1; i <= QUERIES; i++)
   {
            scanf ("query %s %f\n",&qid, &time);
            sum += log (time);
            qts += time ;
            printf ("%3s\t %8.2f\n",qid, time);
   }

   scanf ("UF2 %f\n", &UF2);
   printf ("UF2\t %8.2f\n", UF2);
   qts += UF2 ;
   sum += log(UF2);
   qppd = 3600 * SF * exp (sum/-N);
   qthd = ( SF * 17 * 3600) / qts ;
   printf ("\n==================\n");
   printf ("QppD@%.2f = %.2f\n", SF, qppd);
   printf ("QthD@%.2f = %.2f\n", SF, qthd);
}
```

E-3: postproc.awk

```
BEGIN {output=""}
$1 == "--" && $2 == "QUERY"    {if (output != "")
{close(output);}
                                                output=DIR "/"
$3 ".out" }
output != ""  {print >> output}
```

E-4: UF1.bat

```
if "%1" == "" goto Usage
prTime
dbaccess dssf%1 uf1.sql > uf1.out 2>&1
prTime
goto end

:Usage
echo Usage: uf1 dbsize

:end
```

E-5: uf1.sql

```
begin work;
set pdqpriority 100;
set isolation repeatable read;
create external table orderupd_ext
sameas order
using ( format "delimited",
datafiles ("disk:1:W:\update\order.tbl.u1"),
rejectfile "W:\update\pf1_o_reject%c",
deluxe
);
insert into order
  select * from orderupd_ext;

create external table lineupd_ext
sameas lineitem
using (
format "delimited",
datafiles ("disk:1:W:\update\lineitem.tbl.u1"),
rejectfile "W:\update\pf1_l_reject%c",
deluxe
);
insert into lineitem
  select * from lineupd_ext;

drop table orderupd_ext;
drop table lineupd_ext;
commit work;
```

E-6: uf1_reset.sql

```
set pdqpriority 100;
begin work;
-- Hold the order keys we need to delete
create scratch table orderkey_tmp
(d_orderkey integer)
fragment by hash (d_orderkey) in tempslice1;
-- This table points to order rows that were inserted
create external table orderkey_ext
sameas order
using ( format "delimited",
datafiles ("disk:1:W:\update\order.tbl.u1"),
rejectfile "W:\update\uf1_reset_reject%c",
express
);
insert into orderkey_tmp (d_orderkey)
  select o_orderkey from orderkey_ext order by 1;

delete from order where o_orderkey in
  (select d_orderkey from orderkey_tmp);

delete from lineitem where l_orderkey in
  (select d_orderkey from orderkey_tmp);
drop table orderkey_tmp;
drop table orderkey_ext;
commit work;
```

E-7: UF2.bat

```
if "%1" == "" goto Usage
prTime
dbaccess dssf%1 uf2.sql > uf2.out 2>&1
prTime
goto end

:Usage
```

```
echo Usage: uf2 dbsize

:end
```

E-8: uf2.sql

```
set pdqpriority 100;
set isolation repeatable read;
begin work;
create scratch table okey_tmp
(d_orderkey integer)
fragment by hash (d_orderkey) in tempslice1;
create external table order_delete_ext
sameas okey_tmp
using ( format "delimited",
datafiles ("disk:1:W:\update\delete.0") );
insert into okey_tmp select * from order_delete_ext order by 1;
delete from order where
o_orderkey in
(select d_orderkey from okey_tmp);
delete from lineitem where l_orderkey in (select d_orderkey
from okey_tmp);
drop table okey_tmp;
drop table order_delete_ext;
commit work;
```

E-9: uf2_reset.bat

```
set pdqpriority 100;
set explain on;
begin work;
create external table orderrei_ext
sameas order
using ( format "delimited",
datafiles ("disk:1:W:\update\order_reinsert"),
rejectfile "W:\update\o_reins%c.rej",
deluxe
);
insert into order
   select * from orderrei_ext;
drop table orderrei_ext;

create external table linerei_ext
sameas lineitem
using (
format "delimited",
datafiles ("disk:1:W:\update\line_reinsert"),
rejectfile "W:\update\l_reins%c.rej",
deluxe
);

insert into lineitem
   select * from linerei_ext;

drop table linerei_ext;
commit work;
```

E-10: Start_query

```
begin work;
```

E-11: End_query

```
commit work;
```

E-12: timer.cpp

```
#include <afx.h>
#include <winbase.h>
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

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

        char    q[80];
        DWORD   tickCount, s, ms;
        CTime   time; /*valid data and time from
1/1/1970 to 2/5/2036. */
        int     yy, mon, d, h, min, sec;

        if(argc > 2) {
                printf("Usage: %s [query number]\n",
argv[0]);
        }
        if(argc == 2) strcpy(q, argv[1]);
        else strcpy(q, "");

        tickCount = GetTickCount();
        time = CTime::GetCurrentTime();
        yy = time.GetYear();
        mon = time.GetMonth();
        d = time.GetDay();
        h = time.GetHour();
        min = time.GetMinute();
        sec = time.GetSecond();


        s = tickCount / 1000L;
        ms = tickCount - s*1000L;


        if (argc == 2)
                printf("Query: %s Date: %04d-%02d-
%02d Time: %d:%02d:%02d.%02d  %ld.%03d\n", q, yy, mon,
d, h, min, sec, ms, s, ms);
        else
                printf("Date: %04d-%02d-%02d Time:
%d:%02d:%02d.%02d  %ld.%03d\n", yy, mon, d, h, min, sec,
ms, s, ms);

        exit(0);
}
```

E-13: prTime.cpp

```
#include <afx.h>
#include <winbase.h>
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

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

        CTime   time; /*valid data and time from
1/1/1970 to 2/5/2036. */
        int     yy, mon, d, h, min, sec;


        time = CTime::GetCurrentTime();
        yy = time.GetYear();
```

```
        mon = time.GetMonth();
        d = time.GetDay();
        h = time.GetHour();
        min = time.GetMinute();
        sec = time.GetSecond();


        printf("Date: %2d/%02d   Time: %2d:%2d:%2d\n",
mon, d, h, min, sec);

}
```

Appendix F: ACID Test Source Code

```
/*
 * Sccsid:     @(#)acid.ec              9.1.2.5     8/16/95  20:56:46
 * ACID test implementation for TPC-D
 *
 *
 * this routine acutally does twice the required number of
transactions,
 * one set to alter the data and another to unroll the changes
after things
 * have completed. It relies on semop() for synchronization.
 */

#define DECLARER
/*#define _XOPEN_SOURCE*/
#include "config.h"
#include <stdio.h>
#include <signal.h>
#include <time.h>

#ifdef WIN32
#include <windows.h>
#include <string.h>
#include <fcntl.h>
#include <io.h>
#else
#include <sys/wait.h>
#include <sys/sem.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <unistd.h>
#include <sys/time.h>
#endif /*WIN*/

#include "dss.h"
#include "acid.h"
#ifdef  WIN32
#include "nt.h"
#else
#include "unix.h"
#endif /*WIN32*/

/*
 * status file macros:
 *  SET_FILE(tgt)   -- redirect status messages to tgt, unless
overridden
 *                  on the command line
 *  DEC_DBL         -- convert a dec_t to a dbl; -1 == error
 *  NEW_SUCCESS     -- force the start of a new success file
 *  TIMESTAMP       -- generate a timestamped status message
 */

#define NEW_SUCCESS             sfp = HFILE_ERROR
#define SET_FILE(str)           \
            if ((flags & FL_DEBUG) == 0)    \
               {            \
               if (ofp != NULL) fclose(ofp);    \
               if ((ofp = fopen(str, "a")) == NULL) \
                        {               \
                        fprintf(stderr, "open failed for %s\n",
str); \
                        exit(1);        \
                        }               \
               } \
            else ofp=stdout
#define DEC_DBL(src, tgt) \
            if (dectodbl(&src, &tgt)) \
                tgt = -1
#define TIMESTAMP(note) \
  { \
  time(&t_tmp); \
  fprintf(ofp, "%-40s TIME: %s", note, ctime(&t_tmp)); \
  fflush(ofp); \
  }


/*
 * general defines
 */
#define ORDER_PER_SF            1500000L
//#define STATUS(stmt) if (flags & FL_DEBUG) fprintf stmt;
else fprintf(stderr,"")
#define STATUS(stmt) if (flags & FL_DEBUG) fprintf stmt


$include sqlca;
$include sqlda;
$include decimal;
$include datetime;

$ typedef struct TRAN_T {
            long o;
            int l;
            int d;
            dec_t rprice;
            dec_t qty;
            dec_t tax;
            dec_t disc;
            dec_t eprice;
            dec_t tprice;
            } tran_t;

int c_cnt, t_cnt;
int nprocs;             /* number of active children */
HFILE sfp = HFILE_ERROR;
char keyfile[80] = "";
char logfile[80] = "";
$ char dbname[80];
FILE *ofp = NULL;
int phase2 = 0;
double flt_scale;
time_t t_tmp;
int flags = 0;
$ tran_t work[1000];
long seed;
extern long Seed[];
int undo = 0;                           /* used to turn
undo_transaction on/off */
int child_flg = 0;
char  *spawn_args[25];
int childcnt;
$ char  connName[11][80];

static void process_options(int ac, char **av);
void c_check();
void pick_keys(void);
void parallel_tran(int, int*, int);
void usage(char *);
void post_proc(int);
void do_testa(void);
void do_testc(void);
void do_testi(void);
void do_testd(void);
CHILD_ROUTINE testc_child(CHILD_PARAM);
CHILD_ROUTINE testi_child(CHILD_PARAM);
CHILD_ROUTINE testd_child(CHILD_PARAM);
void do_semop(int, int);
```

```
void build_tran(int);
void init(int);
void wrapup(void);
void do_tran(int, int);
void undo_tran(int);
void prt_history(int);    /* print the history records */
void dump_row(int, char *);          /* print the named rows for
o/l */
long UnifInt PROTO((long, long, long));

main(int ac, char **av)
{
   int i;
   int chnum;
#ifdef WIN32
   tid_primary = GetCurrentThreadId();
   STATUS((stderr, "Thread%d, main(): ac=%d \n",
tid_primary, ac));
#endif

   process_options(ac, av);
  if (child_flg == 0)
  {
    MAKESEMS;
  }
  else
  {
    OPENSEMS;
  }

  chnum = child_flg - 1;

   if (flags & FL_TESTC)
   {
      if (child_flg != 0)
        testc_child(chnum);
              else
                do_testc();
   }
   if (flags & FL_TESTI)
   {
      if (child_flg != 0)
        testi_child(chnum);
              else
              do_testi();
   }
   if (flags & FL_TESTA)
                do_testa();
   if (flags & FL_TESTD)
   {
      if (child_flg != 0)
        testd_child(chnum);
              else
                do_testd();
   }

   ALLGO;                /* to clear anyone who's left */
   DROPSEMS;

   exit(0);
}

void
c_check(int tnum)
{
   double t1, t2;
   $int okey;
   int i = 0;
   $decimal otot,
                  res;
   $begin work;
   $ whenever not found continue;
   $ whenever error call do_error;
   okey = work[tnum].o;
   $execute o_stmt into $otot using $okey;
   if (SQLCODE == SQLNOTFOUND)
              {
              fprintf(ofp, "%ld is not a valid orderkey\n", okey);
              return;;
              }
   $execute l_stmt into $res using $okey;
   if (SQLCODE == SQLNOTFOUND)
              {
              fprintf(ofp, "%ld has no lines!\n", okey);
              return;
              }
   dectodbl(&otot, &t1);
   dectodbl(&res, &t2);
   fprintf(ofp, "%s orderkey #%ld: wanted %8.2f, got %8.2f\n",
              (deccmp(&otot, &res))?"ERROR: Bad":"success
for", okey, t1, t2);

   $commit work;
   return;
}

static void
process_options(int cnt, char **vector)
{
  extern int optind, opterr;
  extern char *optarg;
  int flg,
     i;

  flags |= FL_TESTALL;
  seed = Seed[0];
  while ((flg = getopt(cnt, vector, "C:uDd:hk:l:n:r:s:t:T:K:"))
!= -1)
              {
              switch (flg)
                 {
                 case 'd':
                              strcpy(dbname, optarg);
                              flags |= FL_DBNAME;
                              break;
                 case 'D':
                              flags |= FL_DEBUG;
                              break;
                 case 'C':
                              children = atoi(optarg);
                              flags |= FL_STREAMS;
                              break;
                 case 'h':
                              usage(vector[0]);
                              exit(0);
                              break;
                 case 'k':
                              if (strlen(optarg) > 79)
                                 {
                                 printf("pathname '%s' exceeds 80
character limit\n", optarg);
                                 exit(1);
                                 }
                              strcpy(keyfile, optarg);
                              flags |= FL_KEY;
                              break;
                 case 'l':
                              if (strlen(optarg) > 79)
                                 {
                                 printf("pathname '%s' exceeds 80
character limit\n", optarg);
```

```
                        exit(1);
                        }
                    strcpy(logfile, optarg);
                    flags |= FL_LOG;
                    break;
                case 'n':
                    c_cnt = atoi(optarg);
                    flags |= FL_COUNT;
                    break;
                case 'r':
                    seed = atol(optarg);
                    flags |= FL_SEED;
                    break;
                case 's':
                    flt_scale = atof(optarg);
                    flags |= FL_SCALE;
                    break;
                case 't':
                    t_cnt = atoi(optarg);
                    if (t_cnt >= 1000)
                        {
                        printf("-t upper bound is 1000\n");
                        t_cnt = 999;
                        }
                    flags |= FL_TRANS;
                    break;
                case 'u':
                    undo = 1;
                    break;
                case 'T':
                    flags &= ~FL_TESTALL;
                    if (*optarg == 'A' || *optarg == 'a')
                        flags |= FL_TESTA;
                    if (*optarg == 'C' || *optarg == 'c')
                        flags |= FL_TESTC;
                    if (*optarg == 'I' || *optarg == 'i')
                        flags |= FL_TESTI;
                    if (*optarg == 'D' || *optarg == 'd')
                        flags |= FL_TESTD;
                    break;
                case 'K':
                    child_flg = atoi(optarg);
            break;

                }
            }
    if ((flags & FL_SCALE) == 0)
            flt_scale = 0.1;
    if ((flags & FL_COUNT) == 0)
            c_cnt = 10;
    if ((flags & FL_TRANS) == 0)
            t_cnt = 100;
    if ((flags & FL_STREAMS) == 0)
            children = 1;
    if ((flags & FL_DBNAME) == 0)
            if (flt_scale < 0) sprintf(dbname, "dssfq");
            else sprintf(dbname, "dssf%3.0f", flt_scale);
    if ((flags & FL_SEED) == 0)
            for (i=0; i < DSS_PROC; i++)
                UnifInt(1L, 100L, 0);
#if (defined(WIN32)&&!defined(_POSIX_))
    if(child_flg == 0) {
      for (i=0; i < cnt; i++)
        {
        spawn_args[i] = malloc((strlen(vector[i]) + 1) *
sizeof(char));
        MALLOC_CHECK(spawn_args[i]);
        strcpy(spawn_args[i], vector[i]);
        }
        spawn_args[cnt] = malloc(3 * sizeof(char));
        MALLOC_CHECK(spawn_args[cnt]);
        strcpy(spawn_args[cnt], "-K");

        spawn_args[cnt+1] = malloc(2 * sizeof(char));
        MALLOC_CHECK(spawn_args[cnt+1]);
        strcpy(spawn_args[cnt+1], "0");
      spawn_args[cnt+2] = NULL;
    }
    childcnt = cnt + 1;
#endif
    return;
}

void
pick_keys(void)
{
    int *res, i, j;
    FILE *kfp;

    if (strlen(keyfile) != 0)
            {
            if ((kfp = fopen(keyfile, "r")) == (FILE *)NULL)
                {
                printf("Unable to open key file '%s'\n", keyfile);
                exit(1);
                }
            fscanf(kfp, "%d\n", &c_cnt);
            for (i=0; i < c_cnt; i++)
                fscanf(kfp, "%d\n", work[i].o);
            fclose(kfp);
            }
    else
            {
            res = (int *)malloc(sizeof(int) * (c_cnt + 1));
            MALLOC_CHECK(res);
            for (i=0; i < c_cnt; i++)
                work[i].o =
                        MK_SPARSE(UnifInt(1L,
(long)(ORDER_PER_SF * flt_scale), 0L), 0);
            }

    return;
}

void
usage(char *prog)
{
  printf("USAGE: %s [options]\n\n\tOptions\n\t=======\n",
prog);
    printf("\t-d <name>\t-- run against database <name>\n");
    printf("\t-h\t\t-- generate this usage message\n");
    printf("\t-C <procs>\t-- run <procs> transaction streams\n");
    printf("\t-k <file>\t-- read keys from <file>\n");
    printf("\t-l <file>\t-- use <file> as the OnLine log file\n");
    printf("\t-n <keycount>\t-- use <keycount> random keys for
verification\n");
    printf("\t-s <SF>\t\t-- assume scale factor SF\n");
    printf("\t-t <trans>\t-- issues <trans> transactions per
stream\n");
    printf("\t-u\t\t-- undo any updates to database\n");
    printf("\t-D\t\t-- output resuts to stdout\n");

    return;
}

void
post_proc(c)
$parameter int c;
{
    char n[80];
    int i;
    $int pkey, skey, okey, lkey, delta, child;
```

```
    $dtime_t d_time;

    sprintf(n, "Transactions from stream #%d\n", c);
    TIMESTAMP(n);
    fprintf(ofp, "%-10s|%-2s|%-3s|%-8s|%-6s|%s\n",
                "order", "l", "dlt", "part", "supp", "time");
    $begin work;
    $open h_crsr using :c;
    $fetch h_crsr
                into :pkey, :skey, :okey, :lkey, :delta, :d_time,
:child;
    while (!SQLCODE)
                {
                dttoasc(&d_time, n);
                fprintf(ofp, "%-10d|%-2d|%-3d|%-8d|%-6d|%s\n",
                   okey, lkey, delta, pkey, skey, n);
                $fetch h_crsr
                   into :pkey, :skey, :okey, :lkey, :delta, :d_time,
:child;
                }
    $close h_crsr;
    $commit work;
    fflush(ofp);

    return;
}

void
do_testa(void)
{
    int i;

    init(0);
    undo_tran(-99);

    $delete from history;
    SET_FILE("atomc");
    fprintf(ofp, "ATOMICITY TEST ONE\n");
    dump_row(0, "Initial State:");
    TIMESTAMP("Committed Transaction (History Table)");
    prt_history(0);
    $begin work;
    do_tran(0, 0);
    $commit work;
    dump_row(0, "Final State:");
    TIMESTAMP("Committed Transaction (History Table)");
    prt_history(0);
    /*TIMESTAMP("atomicity/commit success file entries");*/
    system("cat success.0 >> atomc1");
    undo_tran(0);
    NEW_SUCCESS;

    $delete from history;
    SET_FILE("atomr");
    fprintf(ofp, "ATOMICITY TEST TWO\n");
    dump_row(1, "Initial State:");
    TIMESTAMP("Committed Transaction (History Table)");
    prt_history(1);
    $begin work;
    do_tran(1, 1);
    fprintf(ofp, "\n");
    TIMESTAMP("Requesting Rollback");
    $rollback work;
    TIMESTAMP("Rollback Complete");
    fprintf(ofp, "\n");
    dump_row(1, "Final State:");
    TIMESTAMP("Committed Transaction (History Table)");
    prt_history(1);
    /*TIMESTAMP("atomicity/commit success file entries");*/
    system("cat success.1 >> atomr1");

    wrapup();

    return;
}

void
do_testc(void)
{
    int *p, i, c_pid, j, status;
    tran_t *damage;
    char cmd[80];

    nprocs = children;
    for (i = 0; i < children; i++)
       {
              *spawn_args[childcnt]+=1;
       switch (c_pid = SPAWN())
          {
          case -1:
             fprintf(ofp, "Fork failed for child #%d\n", i);
             for (j=0; j < i; j++)
                KILL(pids[j]);
             exit(1);
             break;
          case 0:    /* CHILD */
             testc_child(i);
                          exit(0);
                          break;
                default:
            pids[i] = c_pid;
            break;
          }
       }
              init(0);
    STATUS((stderr, "Parent: do_testc(): init(0) done\n"));

    undo_tran(-99);
              if (children != 0 )
           $delete from history;
              SET_FILE("consb");
              $set isolation to repeatable read;
              TIMESTAMP("Initial State");
              for (i=0; i < c_cnt; i++)
                 c_check(i);

    if (children == 0)
        exit(0);

              SET_FILE("consckpt");
    TIMESTAMP("Allow Transactions");
              ALLREADY;             /* everyone is ready */
              ALLGO;
              sleep(CKPT_WAIT);
              system("onmode -c >> consckpt1");
              system("onstat -m >> consckpt1");
              ALLREADY;             /* everyone is done */

              SET_FILE("consa");
              TIMESTAMP("Final State");
              for (i=0; i < c_cnt; i++)
                 c_check(i);

              for (i=0; i < children; i++)
                 {
                 ONEGO; /* allow them to post process in turn */
                 sleep(5);
                 }
              ALLGO; /* allow them to terminate */
              sleep(5);
              wrapup();
```

```
   return;
}

void
do_testi(void)
{

   double   cost;
   int      delta1, delta2, hold;
   int      i, j, c_pid;

   $char    comment[199];
   $char    stmt_buf1[1000];
   $char    stmt_buf2[1000];
   $decimal supplycost;
   $int     availqty;
   $long    max_partkey, max_suppkey, partkey, suppkey;
   char     buf[1024];

   nprocs = 1; /* these are $all 1 parent / 1 child tests */
/* JMS
   hold = c_cnt;
   c_cnt = 1;
*/
                  *spawn_args[childcnt]+=1;
            switch (c_pid = SPAWN())
            {
            case -1:
                  fprintf(ofp, "Fork failed for child #%d\n", i);
                  for (j=0; j < i; j++)
                        KILL(pids[j]);
                  exit(1);
                  break;
            case 0: /* CHILD */
                  testi_child(i);
                  exit(0);
                  break;
            default: /* PARENT */
                 init(0);

                 /*
                  * iso1: read-only isolation in the face of a
commit
                  */
                 SET_FILE("iso1");
                 $set isolation to committed read;
                 ALLREADY;   /* process one is waiting to
commit */
                 ALLGO;
                 sleep(5);
                 TIMESTAMP("T2: Initiate Read Query");
                 dump_row(0, "T2: Query Complete");

                 /*
                  * iso2: read-only isolation in the face of a
rollback
                  */
                 ALLREADY;         /* make sure child is ready
for test 2 */
                 ALLGO;            /* let him set up */
                 SET_FILE("iso2");
                 $set isolation to committed read;
                 ALLREADY;   /* process one is waiting to
commit */
                 ALLGO;
                 sleep(5);
                 TIMESTAMP("T2: Initiate Read Query");
                 dump_row(1, "T2: Query Complete");

                 /*
                  * iso3: update isolation in the face of a commit
                  */
                 ALLREADY;         /* process 1 is ready for the
test */
                 ALLGO;            /* let him set up */
                 work[4].o = work[2].o;
                 work[4].l = work[2].l;
                 SET_FILE("iso3");
                 $set isolation to repeatable read;
                 ALLREADY;         /* process 1 is ready to
commit */
                 ALLGO;
                 fprintf(ofp, "\n");
                 sprintf(buf, "T2: Initiate Update (%d,%d)",
work[4].o, work[4].l);
                 TIMESTAMP(buf);
                 fprintf(ofp, "\n");
                 $begin work;
                 do_tran(4, 6);
           TIMESTAMP("T2: Requesting Commit");
                 $commit work;
           TIMESTAMP("T2: Commit Complete");
                 dump_row(4, "T2: Final State:");
                 TIMESTAMP("T2: Committed Transaction
(History Table)");
                 prt_history(6);
                 fflush(ofp);

                 /*
                  * iso4: update isolation in the face of a rollback
                  */
                 NEW_SUCCESS;
                 ALLREADY;         /* process 1 is ready for the
test */
                 ALLGO;            /* let him set up */
                 work[5].o = work[3].o;
                 work[5].l = work[3].l;
                 SET_FILE("iso4");
                 $set isolation to repeatable read;
                 sleep(5);
                 ALLREADY;         /* process 1 is ready to
commit */
                 ALLGO;
                 fprintf(ofp, "\n");
                 sprintf(buf, "T2: Initiate Update (%d,%d)",
work[5].o, work[5].l);
                 TIMESTAMP(buf);
                 fprintf(ofp, "\n");
                 $begin work;
                 do_tran(5, 7);
           TIMESTAMP("T2: Requesting Commit");
                 $commit work;
           TIMESTAMP("T2: Commit Complete");
                 dump_row(5, "T2: Final State:");
                 TIMESTAMP("T2: Committed Transaction
(History Table)");
                 prt_history(7);
                 fflush(ofp);

           /*
            * : concurrent read/write transactions against
            *      different tables
            */
           ALLREADY;
           ALLGO;
           SET_FILE("iso5");
           $set isolation to repeatable read;
                 sleep(2);
           TIMESTAMP("T2: Initiate Query");

           $begin work;
           $select max(ps_partkey) into :max_partkey
```

```
    from partsupp where 1 = 1;
  $select max(ps_suppkey) into :max_suppkey
    from partsupp where 1 = 1;
  partkey = UnifInt(1L, max_partkey, 1L);
  suppkey = UnifInt(1L, max_suppkey, 1L);
  $open ps_crsr using :partkey, :suppkey;
  $fetch ps_crsr into
    :partkey, :suppkey, :availqty, :supplycost, :comment;
  while (SQLCODE)
    {
    $close ps_crsr;
    partkey = UnifInt(1L, max_partkey, 1L);
    suppkey = UnifInt(1L, max_suppkey, 1L);
    $open ps_crsr using :partkey, :suppkey;
    $fetch ps_crsr into
      :partkey, :suppkey, :availqty, :supplycost, :comment;
    }
  $close ps_crsr;

  if (dectodbl(&supplycost, &cost))
    cost = -999.99;
  TIMESTAMP("T2: Query Results:");
  fprintf(ofp,"\n%8s |%8s |%5s |%8s |%s\n",
          "p_key", "s_key", "avqty", "cost", "comment");
  fprintf(ofp,"%8d |%8d |%5d |%8.2f |%s\n\n",
          partkey, suppkey, availqty, cost, comment);
  $commit work;
        fflush(ofp);

  ALLREADY; /* process 1 is ready to commit */
  ALLGO;
        ALLREADY;       /* process 1 is complete */
  ALLGO;

  /*
   * iso6: test to see that update transactions are
   *      not delayed indefinitely when run concurrently
   *      with arbitrary read-only queries
   */

  SET_FILE("iso6");

  $set isolation to repeatable read;
        sleep(3);

  delta1 = UnifInt((long) 0, (long) 2159, (long) 0);

  sprintf(stmt_buf1,"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 <= date('1998-12-01') - interval (%d) day (4)
to day \
group by 1, 2 \
order by 1, 2 into temp temp%d;", delta1, delta1);

  TIMESTAMP("T1: Initiating Q1");
  $begin work;
  $execute immediate :stmt_buf1;
  $commit work;
  TIMESTAMP("T1: Q1 Complete");

  ALLREADY;
  ALLGO;

  delta2 = UnifInt((long) 0, (long) 2159, (long) 0);
  while (delta1 == delta2)
    delta2 = UnifInt((long) 0, (long) 2159, (long) 0);

  sprintf(stmt_buf2,"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 <= date('1998-12-01') - interval (%d) day (4)
to day \
group by 1, 2 \
order by 1, 2 into temp temp%d;", delta2, delta2);

  TIMESTAMP("T1: Initiating Q1");
  $begin work;
  $execute immediate :stmt_buf2;
  $commit work;
  TIMESTAMP("T1: Q1 Complete");
 /* dump_row(7, "Final State:");
        TIMESTAMP("Committed Transaction (History
Table)");
  prt_history(9); */

  ALLREADY;
        ALLGO;    /* allow him to cleanup */
        sleep(10);

        wrapup();
        break;
      }
/* JMS
    c_cnt = hold;
*/
    return;
}

void
do_testd(void)
{
  int c_pid, o, l, d;
#ifndef WIN32
  int pids[MAX_CHILDREN];
#endif /*WIN32*/
  int i, count = 0;

  nprocs = children;
  for (i=0; i < children; i++)
  {
        *spawn_args[childcnt]+=1;
    switch(c_pid = SPAWN())
      {
      case -1:
        perror("fork error: durability");
        exit(1);
        break;
      case 0:
  /* child process/thread runs child routine */
            testd_child(i);
            exit(0);
        break;
      default:
```

```
            break;
          }
    }
    /* parent process/thread continues here */
    init(0);
    undo_tran(-99);
    $delete from history;
    ALLREADY;
    ALLGO;
    exit(0);
}

#ifndef WIN32
void
do_semop(int s, int v)
{
    struct sembuf sop;

    sop.sem_num = s;
    sop.sem_op  = v;
    sop.sem_flg = 0;

    if (semop(semid, &sop, 1) == -1)
        {
        perror("tpcd: semop");
        exit(1);
        }
    return;
}
#endif /*WIN32*/

void
build_tran(int cnt)
{
    int i, good_key;
    static int init = 0;
    $int lnum;


    for (i = 0; i < cnt; i++)
        {
        good_key = 0;
        if (i >= c_cnt)
            work[i].o =
                MK_SPARSE(UnifInt(1L, (long)(ORDER_PER_SF
* flt_scale), 0L), 0);
        while (!good_key)
          {

          $execute get_lnum into :lnum using :work[i].o;
          if (lnum >= 0)
            good_key = 1;
          else
            work[i].o =
                MK_SPARSE(UnifInt(1L, (long)(ORDER_PER_SF
* flt_scale), 0L), 0);
          }
        work[i].l= UnifInt(O_LCNT_MIN, (long)lnum, 0L);
        work[i].d= UnifInt((long)1, (long)100, (long)0);
        }
    return;
}

void
wrapup(void)
{

    $delete from history;

#ifdef WIN32
    $disconnect $connName[PARENTCONN];
#else
    $close database;
    $free get_lnum;
    $free l_stmt;
    $free o_stmt;
    $free h_stmt;
    free h_crsr;
    $free ps_stmt;
    if (flags & FL_TESTI)
      free ps_crsr;
#endif

    return;
}

void
init(childnum)
int childnum;
{
    DWORD   tid;
#ifdef WIN32
    int     idx;
    tid = GetCurrentThreadId();
    if(tid == tid_primary)
      idx = PARENTCONN;
    else
    {
      idx = 0;
      while (tid != dwChildId[idx] && idx <10) idx++;
      if (idx == 10)
      {
        printf("Thread #%d, init(): Error ThreadId not
matched!!\n", tid);
        exit(-1);
      }
    }
    sprintf(connName[idx], "conn%d", tid);

    $connect to $dbname as $connName[idx];

    STATUS((stderr, "Thread #%d Connected to Database %s
as %s\n",
                        tid, dbname, connName[idx]));
#else
    $database $dbname;
        tid = childnum;
#endif /*WIN32*/

    $set pdqpriority 100;
    $whenever error call do_error;
    $set lock mode to wait;
    $set isolation to repeatable read;
    $prepare get_lnum from
        "select max(l_linenumber) from lineitem where
l_orderkey = ?";
/*
 * since the underlying calculation for DBGEN relies on integer
math and money
 * expressed in pennies, while the schema requires a more
standard dollars and
 * cents representation, it is necessary to apply the truncation
that integer
 * math enforces, rather than the rounding that is informix's
default behavior
 */
        STATUS((stderr, "ThreadId: #%d, get_lnum
prepared\n", tid));
    $prepare l_stmt from
      "select sum( trunc( trunc(l_extendedprice * (1 -
l_discount),2) * (1 + l_tax), 2)) from lineitem where
l_orderkey = ?";
```

```
            STATUS((stderr, "ThreadId: #%d, l_stmt
prepared\n", tid));
   $prepare o_stmt from
                        "select o_totalprice from order where
o_orderkey = ?";
            STATUS((stderr, "ThreadId: #%d, o_stmt
prepared\n", tid));
   $prepare h_stmt from
                        "select * from history where h_child = ?
order by h_date";
            STATUS((stderr, "ThreadId: #%d, h_stmt
prepared\n", tid));

   $declare h_crsr cursor for h_stmt;
            STATUS((stderr, "ThreadId: #%d, h_crsr
prepared\n", tid));

   $prepare ps_stmt from
      "select * from partsupp where ps_partkey = ? and
ps_suppkey = ?";

            STATUS((stderr, "ThreadId: #%d, ps_stmt
prepared\n", tid));
   $declare ps_crsr cursor for ps_stmt;

   /* Seed[0] = seed; */
            STATUS((stderr, "ThreadId: #%d, Setting Seed\n",
tid));
/* work around for Francois 7-22
   Seed[0] = seed + ( childnum * 201 );
*/
   Seed[0] = seed;
            STATUS((stderr, "ThreadId: #%d, Entering
pick_keys()\n", tid));
   pick_keys();
            STATUS((stderr, "ThreadId: #%d, Entering
build_tran()\n", tid));
   $begin work;
   build_tran(t_cnt);
   $commit work;
            STATUS((stderr, "ThreadId: #%d, build_tran()
done\n", tid));
   return;
}

void
do_transaction(work_unit, c)
$tran_t *work_unit;
$int c;

{

  $decimal cost, disc, e, new_ototal, otot, ototal_delta, q, rprice,
tax;
  $decimal num1, result1, result2, result3;
  $datetime year to fraction cur_dt;
  $int     pkey, skey;

  $set isolation to repeatable read;
  if (flags & FL_DEBUG)
                $set explain on;

  $select o_totalprice into :otot from order
     where o_orderkey = :work_unit->o;

  $select l_quantity, l_extendedprice, l_partkey, l_suppkey,
l_tax, l_discount
         into :q, :e, :pkey, :skey, :tax, :disc from lineitem
     where l_orderkey = :work_unit->o and l_linenumber =
:work_unit->l;

  //STATUS((stderr, "ThreadId: #%d, Start trans\n", c));
  //Sleep(40);

  deccvint(1, &num1);

  /* 1 - disc */
  decsub(&num1, &disc, &result1);

  /* e times (1-disc) */
  decmul(&e, &result1, &result2);

  /* trunc (e times (1-disc)) */
  dectrunc(&result2, 2);

  /* 1 + tax */
  decadd(&num1, &tax, &result1);

  /* (e times (1-disc)) times (1+tax) */
  decmul(&result2, &result1, &result3);

  /* trunc (e times (1-disc)) times (1+tax) */
  dectrunc(&result3, 2);

  /* otot - ((e times (1-disc)) times (1+tax)) */
  decsub(&otot, &result3, &work_unit->tprice);

  /* e divided by q */
  decdiv(&e, &q, &work_unit->rprice);

  /* trunc (e divided by q) */
  dectrunc(&work_unit->rprice, 2);

  deccvint(work_unit->d, &num1);

  /* d times rprice */
  decmul(&num1, &work_unit->rprice, &cost);

  /* trunc (d times rprice) */
  dectrunc(&cost, 2);

  if (flags & FL_DEBUG)
            {
            char buf[1024];
            sprintf(buf, "(%d) Updating Lineitem (key =
%d,%d)",
                        c, work_unit->o, work_unit->l);
            TIMESTAMP(buf);
            }

  $update lineitem
    set (l_extendedprice, l_quantity) =
       (:e + :cost, :q + :work_unit->d)
       where l_orderkey = :work_unit->o and l_linenumber =
:work_unit->l;

  if (flags & FL_DEBUG)
            {
            char buf[1024];
            sprintf(buf, "(%d) Done Updating Lineitem", c);
            TIMESTAMP(buf);
            }

  if (flags & FL_DEBUG)
            {
            char buf[1024];
            sprintf(buf, "(%d) Updating Order (key = %d)", c,
work_unit->o);
            TIMESTAMP(buf);
            }
```

```
  $update order
    set o_totalprice =
       :work_unit->tprice + trunc(trunc((:e+:cost)*(1-
:disc),2)*(1+:tax),2)
       where o_orderkey = :work_unit->o;

  if (flags & FL_DEBUG)
                {
                char buf[1024];
                sprintf(buf, "(%d) Done Updating Order", c);
                TIMESTAMP(buf);
                }

  $select o_totalprice into :new_ototal from order
    where o_orderkey = :work_unit->o;

  decsub(&new_ototal, &otot, &ototal_delta);

  dtcurrent(&cur_dt);
  if (work_unit->d > 0) {
    $insert into history values
     (:pkey, :skey, :work_unit->o, :work_unit->l, :work_unit-
>d,
      :cur_dt, :cost, :ototal_delta, :c);
                }

}

/*
 * transaction specifics:
 *  do_tran(n, mode, child) -- execute (mode == 0) or undo
(mode == 1) the
 *                          n-th transaction
 */
void
do_tran(n, child)
int n;
$parameter int child;
{

    double r, q, t, d, e, x;
    char name[20];
    int good_key = 0;
    $ int l_num;
    char buf[512];


    if (sfp == -1)
                {
                sprintf(name, "success.%d", child);
                //sfp = _open(name,
_O_CREAT|_O_WRONLY|O_BINARY);
                sfp = _lcreat(name, 0);
                sprintf(buf,"%-10s|%-2s|%-3s|%-10s|%-4s|%-4s|%-
4s|%-10s|%-10s\n",
                   "order", "l", "dlt", "rprice", "qty", "tax",
                   "dsc", "eprice", "tprice");
        _lwrite(sfp, buf, strlen(buf));
                FlushFileBuffers(sfp);
                }

    do_transaction(&work[n], child);

    DEC_DBL(work[n].rprice, r);
    DEC_DBL(work[n].qty, q);
    DEC_DBL(work[n].tax, t);
    DEC_DBL(work[n].disc, d);
    DEC_DBL(work[n].eprice, e);
    DEC_DBL(work[n].tprice, x);

sprintf(buf,"%10ld|%2ld|%3ld|%10.2f|%4.0f|%4.2f|%4.2f|%10.
2f|%10.2f\n",
                work[n].o, work[n].l, work[n].d, r, q, t, d, e, x);
    _lwrite(sfp, buf, strlen(buf));
    FlushFileBuffers(sfp);
    sleep(1);

    return;
}

void
undo_tran(child)
$parameter int child;

{

  $decimal cost, ototal_delta;
  $dtime_t h_date;
  $int     p_key, s_key, o_key, l_key, delta, h_child;

  if (undo == 0) {
                return;
  }

  /*$begin work;*/

  if (child >= 0)
    {
    $open h_crsr using :child;
    $fetch h_crsr into
      $p_key, $s_key, $o_key, $l_key, $delta, $h_date,
      $cost, $ototal_delta, $h_child;
    }
  else
    {
    $prepare hall_stmt from
      "select * from history";
    $declare hall_crsr cursor for hall_stmt;
    $open hall_crsr;
    $fetch hall_crsr into
      $p_key, $s_key, $o_key, $l_key, $delta, $h_date,
      $cost, $ototal_delta, $h_child;
    }

  while (!SQLCODE)
    {
    $update lineitem
      set (l_extendedprice, l_quantity) =
        (l_extendedprice - :cost, l_quantity - :delta)
        where l_orderkey = :o_key and l_linenumber = :l_key;

    $update order
      set (o_totalprice) =
         (o_totalprice - :ototal_delta)
         where o_orderkey = :o_key;

    if (child >= 0)
      $fetch h_crsr into
        $p_key, $s_key, $o_key, $l_key, $delta, $h_date,
        $cost, $ototal_delta, $h_child;
    else
      $fetch hall_crsr into
        $p_key, $s_key, $o_key, $l_key, $delta, $h_date,
        $cost, $ototal_delta, $h_child;
    }

    if (child >= 0)
      $close h_crsr;
    else
      $close hall_crsr;

    /*$commit work;*/
```

```
 return;

}

void
prt_history(h)
$parameter int h;
{
   $decimal cost, ototal_delta;
   $int p_key, s_key, o_key, l_key, delta, h_child;
   $dtime_t h_date;
   char dstr[40];

   $begin work;
   $open h_crsr using :h;
   $fetch h_crsr into
            $p_key, $s_key, $o_key, $l_key, $delta, $h_date,
      $h_child;

   if (!SQLCODE) {
            fprintf(ofp, "\n\t%8s |%8s |%9s |%2s |%3s |%s\n",
                     "p_key", "s_key", "o_key", "l", "d", "
date");
   } else {
            fprintf(ofp, "\n\tNo rows returned\n");
   }

   while (!SQLCODE)
            {

            if (dttoasc(&h_date, dstr))
               strcpy(dstr, "error");

      fprintf(ofp, "\t%8d |%8d |%9d |%2d |%3d | %s\n",
         p_key, s_key, o_key, l_key, delta, dstr);

      $fetch h_crsr into
         $p_key, $s_key, $o_key, $l_key, $delta, $h_date,
         $h_child;
            }

   $close h_crsr;
   $commit work;

   fprintf(ofp, "\n");
   fflush(ofp);

   return;
}

void
dump_row(tnum, which_tran)
            int tnum;
            char *which_tran;
{
   $int linenumber, okey;
   $dec_t qty, eprice, otot;
   static int init = 0;
   double d;
   char msgbuf[40];

   if (init == 0)
            {
            $prepare d_stmt from
            "select l_quantity, l_extendedprice, l_linenumber
from lineitem where l_orderkey = ? order by l_linenumber";
            $declare d_crsr cursor for d_stmt;
      //init = 1;
            }
   okey = work[tnum].o;
   STATUS((stderr, "\n\trandomly selected order key =
%d\n\n", okey));
   fprintf(ofp, "\n");
   $open d_crsr using $okey;
   STATUS((stderr, "\n\d_crsr opened for order key = %d\n\n",
okey));
   $fetch d_crsr into $qty, $eprice, $linenumber;
   STATUS((stderr, "\n\d_crsr fetched for order key = %d\n\n",
okey));

   TIMESTAMP(which_tran);

   fprintf(ofp, "\n\t%8s |%10s |%11s |\n", "line", "quantity", "ext
price");

   while (!SQLCODE)
            {
            if (dectodbl(&qty, &d))
               d = -1;

            if (work[tnum].l == linenumber) {
                     fprintf(ofp, "\t*%7d |%10.2f |",
linenumber, d);
            } else {
                     fprintf(ofp, "\t%8d |%10.2f |",
linenumber, d);
            }
            if (dectodbl(&eprice, &d))
               d = -1;
            fprintf(ofp, "%11.2f |\n", d);
            $fetch d_crsr into $qty, $eprice, $linenumber;
            }
   $execute o_stmt into $otot using $okey;
   if (dectodbl(&otot, &d))
            d = -1;
   fprintf(ofp, "\n\t%8s %10.2f\n\n", "Total:", d);
   fflush(ofp);
   $close d_crsr;
   $free d_crsr;
   $free d_stsmt;


   return;
}

CHILD_ROUTINE testc_child(CHILD_PARAM lpPara)
{
            int i, j;
            DWORD tid;
            tid = lpPara;
            STATUS((stderr, "testc_child: child started,
ThreadId: #%d\n", tid));

            i = (int) lpPara;
            SET_FILE("stderr");
            init(0);
            STATUS((stderr, "testc_chilld, ThreadId: #%d,
init0 done\n", tid));

            NEW_SUCCESS;
            $set isolation to repeatable read;
            CHILDSYNC;
            STATUS((stderr, "testc_chilld, ThreadId: #%d,
starting trans \n", tid));
            for (j=0; j < t_cnt; j++)
         if( ((j+1) % children) == i)
                              {
                              $begin work;
                              $set lock mode to wait;
                              do_tran(j, i);
```

```
                                        $commit work;
                                        }
                STATUS((stderr, "testc_chilld, ThreadId: #%d, end
trans \n", tid));
                CHILDSYNC;
                SET_FILE("consrte");
                post_proc(i);
                undo_tran(i);
                CHILDSYNC;
                return(0);
}

CHILD_ROUTINE testi_child(CHILD_PARAM lpPara)
{
   char     buf[1024];
                int i;

                i = (int) lpPara;

                init(0);
                undo_tran(-99);
                $delete from history;

                /*
                 * iso1: read-only isolation in the face of a commit
                 */
                SET_FILE("iso1");
                fprintf(ofp, "ISOLATION TEST ONE\n");
                $set isolation to repeatable read;
                dump_row(0, "Initial State:");
                TIMESTAMP("History table Contents:");
                prt_history(2);

                sprintf(buf, "T1: Initiate Update (%d,%d)",
work[0].o, work[0].l);
                TIMESTAMP(buf);

                $begin work;
                do_tran(0, 2);
                TIMESTAMP("T1: Suspending");
                fprintf(ofp, "\n");
                CHILDSYNC;
                sleep(20); /* sleep to be *sure* there is overlap */
                TIMESTAMP("T1: Requesting Commit");
                $commit work;
                TIMESTAMP("T1: Commit Done");
                fprintf(ofp, "\n");
                sleep(5); /* sleep to be *sure* trace file is obvious */

                dump_row(0, "Final State:");
                TIMESTAMP("Committed Transaction (History
Table)");
                prt_history(2);
                undo_tran(2);
                CHILDSYNC;
                fflush(ofp);

                /*
                 * iso2: read-only isolation in the face of a rollback
                 */
                $delete from history;
                NEW_SUCCESS;
                SET_FILE("iso2");
                fprintf(ofp, "
```

```
ISOLATION TEST TWO\n");
            $set isolation to repeatable read;
            dump_row(1, "Initial State:");
            TIMESTAMP("Committed Transaction (History
Table)");
            prt_history(3);
            sprintf(buf, "T1: Initiate Update (%d,%d)",
work[1].o, work[1].l);
            TIMESTAMP(buf);
            $begin work;
            do_tran(1, 3);
            TIMESTAMP("T1: Suspending");
            CHILDSYNC;
            sleep(10); /* sleep to be *sure* there is overlap */
            TIMESTAMP("T1: Requesting Rollback");
            $rollback work;
            TIMESTAMP("T1: Rollback Complete");
            sleep(10); /* sleep to be *sure* trace file is obvious
*/
            dump_row(1, "Final State:");
            TIMESTAMP("Committed Transaction (History
Table)");
            prt_history(3);
            CHILDSYNC;
            fflush(ofp);

            /*
             * iso3: update isolation in the face of a commit
             */
            $delete from history;
            NEW_SUCCESS;
            SET_FILE("iso3");
            fprintf(ofp, "
```

```
ISOLATION TEST THREE\n");
            $set isolation to repeatable read;
            dump_row(2, "Initial State");
            TIMESTAMP("Committed Transaction (History
Table)");
            prt_history(4);
            sprintf(buf, "T1: Initiate Update (%d,%d)",
work[2].o, work[2].l);
            TIMESTAMP(buf);
            $begin work;
            do_tran(2, 4);
            TIMESTAMP("T1: Suspending");
            CHILDSYNC;
            sleep(10); /* sleep to be *sure* there is overlap */
            TIMESTAMP("T1: Requesting Commit");
            $commit work;
            TIMESTAMP("T1: Commit Complete");
            fprintf(ofp, "\n");
            /* sleep(10);  sleep to be *sure* trace file is obvious
*/
            dump_row(2, "T1: Final State:");
            TIMESTAMP("T1: Committed Transaction
(History Table)");
            prt_history(4);
            CHILDSYNC;
            fflush(ofp);

            /*
             * iso4: update isolation in the face of a rollback
             */
            $delete from history;
            NEW_SUCCESS;
            SET_FILE("iso4");
            fprintf(ofp, "
```

```
ISOLATION TEST FOUR\n");
            $set isolation to repeatable read;
            dump_row(3, "Initial State");
            TIMESTAMP("Committed Transaction (History
Table)");
            prt_history(5);
            sprintf(buf, "T1: Initiate Update (%d,%d)",
work[3].o, work[3].l);
            TIMESTAMP(buf);
            $begin work;
            do_tran(3, 5);
            TIMESTAMP("T1: Suspending");
            CHILDSYNC;
            sleep(10); /* sleep to be *sure* there is overlap */
            TIMESTAMP("T1: Requesting Rollback");
            $rollback work;
            TIMESTAMP("T1: Rollback Complete");
            /* sleep(10);  sleep to be *sure* trace file is obvious
*/
            dump_row(3, "T1: Final State:");
            TIMESTAMP("T1: Committed Transaction
(History Table)");
            prt_history(5);
            fflush(ofp);
            CHILDSYNC;

            /*
             * iso5: concurrent read/write transactions against
             *       different tables
             */
            $delete from history;
            NEW_SUCCESS;
            SET_FILE("iso5");
            fprintf(ofp, "
```

```
ISOLATION TEST FIVE\n");
            $set isolation to repeatable read;
            dump_row(6, "Initial State");
            TIMESTAMP("Committed Transaction (History
Table)");
            prt_history(8);
            sprintf(buf, "T1: Initiate Update (%d,%d)",
work[6].o, work[6].l);
            TIMESTAMP(buf);
            $begin work;
            do_tran(6,8);
            TIMESTAMP("T1: waiting to commit");
            CHILDSYNC;
            TIMESTAMP("T1: Requesting Commit");
            $commit work;
            TIMESTAMP("T1: Commit Complete");
            dump_row(6, "Final State:");
            TIMESTAMP("Committed Transaction (History
Table)");
            prt_history(8);
            fflush(ofp);
            CHILDSYNC;

            /*
             * iso6: test to see that update transactions are
             *      not delayed indefinitely when run
concurrently
             *      with arbitrary read-only queries
             */
            NEW_SUCCESS;
            SET_FILE("iso6");
            fprintf(ofp, "
```

```
ISOLATION TEST SIX\n");
        fflush(ofp);
        $set isolation to repeatable read;
        dump_row(7, "Initial State");
        TIMESTAMP("Committed Transaction (History
Table)");
        prt_history(9);
        sleep(3);
        sprintf(buf, "T2: Initiate Update (%d,%d)",
work[7].o, work[7].l);
        TIMESTAMP(buf);
        $begin work;
        do_tran(7,9);
        CHILDSYNC;
        sleep(3);
        TIMESTAMP("T2: Requesting Commit");
        $commit work;
        TIMESTAMP("T2: Commit Complete");
        dump_row(7, "Final State:");
        TIMESTAMP("Committed Transaction (History
Table)");
        prt_history(9);
        CHILDSYNC;
        fflush(ofp);

        return(0);
}

CHILD_ROUTINE testd_child(CHILD_PARAM lpPara)
{
        int i, count = 0;
        nprocs = children;
        i = (int) lpPara;

        sleep(i);
        init(i);
        SET_FILE("durrate");
        CHILDSYNC;
        while(1)
                {
                TIMESTAMP("Begin Transaction");
                $begin work;
                do_tran(count % t_cnt, i);
                $commit work;
                TIMESTAMP("End Transaction");
                if (++count % 100 == 0)
                        {
                        char msg[60];
                        sprintf(msg, "Stream %d:
Batch of 100 transactions complete", i);
                        TIMESTAMP(msg);
                        }
                if (count % t_cnt == 0)
                        {
                        char msg[60];
                        sprintf(msg, "Stream %d:
%d transactions complet e", i, t_cnt);
                        TIMESTAMP(msg);
                        build_tran(t_cnt);
                        }
                }
        return(0);
}
```

F-2: nt.h

```
/*
 * Synchroniztion macros:
 * MAKESEMS()  -- create the semaphores
 * ALLREADY()  -- delete the semaphores
 *
 * synchronization scheme:
 *    sem[0] -- number of children ready to go
 *    sem[1] -- number of children allowed to go
 *    sem[2] -- ???
 * CHILDSYNC() -- this child registers as ready, then waits for
permission,
 *              to continue
 * ALLREADY()  -- parent will wait here until all children are
ready
 * ALLGO()     -- parent allows all children to proceed
 * ONEGO()     -- parent allows *a* child to proceed
 */
#define MAKESEMS \
      if(( semid[0] = CreateSemaphore(NULL,0,10,"sem-0"))
== NULL) \
            fprintf(stderr, "CreateSemaphore-0 FAILS\n"); \
      if(( semid[1] = CreateSemaphore(NULL,0,10,"sem-1"))
== NULL) \
            fprintf(stderr, "CreateSemaphore-1 FAILS\n"); \
      if(( semid[2] = CreateSemaphore(NULL,1,10,"sem-2"))
== NULL) \
            fprintf(stderr, "CreateSemaphore-2 FAILS\n");
#define DROPSEMS \
              CloseHandle(semid[0]); \
              CloseHandle(semid[1]); \
              CloseHandle(semid[2])
#define CHILDSYNC \
      ReleaseSemaphore(semid[0], 1, NULL); \
      WaitForSingleObject(semid[1], INFINITE)
#define ALLREADY \
      for(sem_index=0; sem_index < nprocs; sem_index++) \
            WaitForSingleObject(semid[0], INFINITE)
#define ALLGO ReleaseSemaphore(semid[1], nprocs, NULL)
#define ONEGO ReleaseSemaphore(semid[1], 1, NULL)

/*
 * datatype defines to map routines/objects to and from
NT/Unix
 *
 */
#define THREAD_DECL __declspec(thread)
#define PID int
#define CHILD_ROUTINE DWORD WINAPI
#define CHILD_PARAM LPVOID

/*
 * process creation/destruction macros
 *
 * CHILD_START(<routine>) creates <nprocs>
processes/threads to
 *                    execute <routine>
 * KILL_CHILD  clean up after the child is done
 */
#define KILL_CHILD for (i = 0; i < nprocs; i++)
CloseHandle(pids[i])
#define CHILD_START(routine) \
      for (i = 0; i < nprocs; i++) \
                                    { \
                                    threadPara = i; \

            pids[i]=CreateThread(NULL,0,routine,threadPara,
0, &dwChildId[i]); \
                                    if(pids[i]==NULL) \
            { \
                                          lastError =
GetLastError(); \
                                          fprintf(ofp,
"CreateThread Error: %d, Instance: %d\n", \

            lastError, i); \
```

```
                                                exit(1); \
        } \
                                        }

/*
 * miscelaneous defines and declarations
 */
#define LOCKCONNNAME WaitForSingleObject(semid[2],
INFINITE)
#define UNLOCKCONNNAME ReleaseSemaphore(semid[2],
1, NULL)
#define sleep(s) Sleep(s*1000)
#define PARENTCONN 11
int           sem_index;
HANDLE  semid[3]; /* timing control semaphores */
HANDLE  pids[MAX_CHILDREN];
DWORD   dwChildId[10];
DWORD   tid_primary;
DWORD   lastError;
int     threadPara;
$ char  connName[11][80];
```

Appendix G: Database Contents

G-1: Lineitem contents

l_orderkey 1
l_partkey 4655681
l_suppkey 230697
l_linenumber 1
l_quantity 17.00
l_extendedprice 27819.65
l_discount 0.04
l_tax 0.02
l_returnflag N
l_linestatus O
l_shipdate 1996-03-13
l_commitdate 1996-02-12
l_receiptdate 1996-03-22
l_shipinstruct DELIVER IN PERSON
l_shipmode TRUCK
l_comment iPBw4mMm7w7kQ zNPL i261OPP

l_orderkey 1
l_partkey 2019273
l_suppkey 219274
l_linenumber 2
l_quantity 36.00
l_extendedprice 42918.12
l_discount 0.09
l_tax 0.06
l_returnflag N
l_linestatus O
l_shipdate 1996-04-12
l_commitdate 1996-02-28
l_receiptdate 1996-04-20
l_shipinstruct TAKE BACK RETURN
l_shipmode MAIL
l_comment 5wM04SNyl0AnghCP2nx lAi

l_orderkey 1
l_partkey 1910994
l_suppkey 110995
l_linenumber 3
l_quantity 8.00
l_extendedprice 16039.20
l_discount 0.10
l_tax 0.02
l_returnflag N
l_linestatus O
l_shipdate 1996-01-29
l_commitdate 1996-03-05
l_receiptdate 1996-01-31
l_shipinstruct TAKE BACK RETURN
l_shipmode REG AIR
l_comment SQC2C 5PNCy4mM

l_orderkey 1
l_partkey 63945
l_suppkey 138946
l_linenumber 4
l_quantity 28.00
l_extendedprice 53450.32
l_discount 0.09
l_tax 0.06
l_returnflag N
l_linestatus O
l_shipdate 1996-04-21
l_commitdate 1996-03-30
l_receiptdate 1996-05-16
l_shipinstruct NONE
l_shipmode AIR
l_comment Om0L65CSAwSj5k6k

l_orderkey 1
l_partkey 720800
l_suppkey 45807
l_linenumber 5
l_quantity 24.00
l_extendedprice 43698.48
l_discount 0.10
l_tax 0.04
l_returnflag N
l_linestatus O
l_shipdate 1996-03-30
l_commitdate 1996-03-14
l_receiptdate 1996-04-01
l_shipinstruct NONE
l_shipmode FOB
l_comment C2gOQj OB6RLk1BS15 igN

l_orderkey 1
l_partkey 469034
l_suppkey 19037
l_linenumber 6
l_quantity 32.00
l_extendedprice 32096.32
l_discount 0.07
l_tax 0.02
l_returnflag N
l_linestatus O
l_shipdate 1996-01-30
l_commitdate 1996-02-07
l_receiptdate 1996-02-03
l_shipinstruct DELIVER IN PERSON
l_shipmode MAIL
l_comment CB0SnyOL PQ32B70wB75k 6Aw10m0wh

l_orderkey 2
l_partkey 5045588
l_suppkey 245589
l_linenumber 1
l_quantity 24.00
l_extendedprice 36799.92
l_discount 0.00
l_tax 0.08
l_returnflag N
l_linestatus O
l_shipdate 1997-03-05
l_commitdate 1997-02-09
l_receiptdate 1997-03-11
l_shipinstruct COLLECT COD
l_shipmode AIR
l_comment O52M70MRgRNnmm476mNm

l_orderkey 3
l_partkey 3185092
l_suppkey 35113
l_linenumber 1
l_quantity 38.00
l_extendedprice 44723.72
l_discount 0.00
l_tax 0.05
l_returnflag A
l_linestatus F
l_shipdate 1993-12-11
l_commitdate 1993-11-27
l_receiptdate 1993-12-16
l_shipinstruct TAKE BACK RETURN

l_shipmode RAIL
l_comment 3AR yMS77lQ12kR

l_orderkey 3
l_partkey 5835248
l_suppkey 210268
l_linenumber 2
l_quantity 30.00
l_extendedprice 35488.50
l_discount 0.05
l_tax 0.01
l_returnflag A
l_linestatus F
l_shipdate 1994-02-05
l_commitdate 1993-12-29
l_receiptdate 1994-02-18
l_shipinstruct TAKE BACK RETURN
l_shipmode FOB
l_comment 6wQnO0Llg6y

l_orderkey 3
l_partkey 3004902
l_suppkey 4903
l_linenumber 3
l_quantity 44.00
l_extendedprice 79497.00
l_discount 0.07
l_tax 0.03
l_returnflag R
l_linestatus F
l_shipdate 1994-01-19
l_commitdate 1993-12-23
l_receiptdate 1994-02-15
l_shipinstruct TAKE BACK RETURN
l_shipmode SHIP
l_comment LhiA7wygz0k4g4zRhMLBAM

10 row(s) retrieved.

G-2: Order contents

o_orderkey 1
o_custkey 2340049
o_orderstatus O
o_totalprice 206270.27
o_orderdate 1996-01-02
o_orderpriority 5-LOW
o_clerk Clerk#000028517
o_shippriority 0
o_comment A0xCm5ARNL
mxjChn2kC64xA4L6zBg2O5jhg M42izyPO
QlymN1ky5kmSiSg
BAQA

o_orderkey 2
o_custkey 3699418
o_orderstatus O
o_totalprice 39743.91
o_orderdate 1996-12-01
o_orderpriority 1-URGENT
o_clerk Clerk#000026375
o_shippriority 0
o_comment 5PRxL1nM7xhQNzP2hnjhy1zz ykhg4P2A
MMg5Px3OCN 0B0iyCRgiC2

o_orderkey 3
o_custkey 4103281
o_orderstatus F
o_totalprice 252918.76
o_orderdate 1993-10-14
o_orderpriority 5-LOW
o_clerk Clerk#000028628
o_shippriority 0
o_comment nm0kygQBnw7RS3AAA4k

o_orderkey 4
o_custkey 1668661
o_orderstatus O
o_totalprice 7119.05
o_orderdate 1995-10-11
o_orderpriority 5-LOW
o_clerk Clerk#000003702
o_shippriority 0
o_comment CP42CySQlz64n3mCyjm17 4B0CL
L5772m4k2Ai4h1nPySwSmNyCl4jOAOx5y4
Rjx36nhO1x2x4Qw

o_orderkey 5
o_custkey 3901709
o_orderstatus F
o_totalprice 170010.35
o_orderdate 1994-07-30
o_orderpriority 5-LOW
o_clerk Clerk#000027744
o_shippriority 0
o_comment 3PNC7zMP534MSizgy34Bxj62l0C7n6PBk7

o_orderkey 6
o_custkey 1830002
o_orderstatus F
o_totalprice 2876.06
o_orderdate 1992-02-21
o_orderpriority 4-NOT SPECIFIED
o_clerk Clerk#000001740
o_shippriority 0
o_comment 1CN00NA0z75SwwCxMNB0MLNL

o_orderkey 7
o_custkey 3827629
o_orderstatus O
o_totalprice 201161.38
o_orderdate 1996-01-10
o_orderpriority 2-HIGH
o_clerk Clerk#000014089
o_shippriority 0
o_comment gmiC6hj5L4 0ixCAQkmB6giC16l4L16g

o_orderkey 32
o_custkey 3457553
o_orderstatus O
o_totalprice 258512.05
o_orderdate 1995-07-16
o_orderpriority 2-HIGH
o_clerk Clerk#000018469
o_shippriority 0
o_comment 7ihNSz00NCxA31PPx6RM4ih
BPPlz417SLk3SRA1zx0nlikRgjkx

o_orderkey 33
o_custkey 2583433
o_orderstatus F
o_totalprice 78202.50
o_orderdate 1993-10-27
o_orderpriority 3-MEDIUM
o_clerk Clerk#000012258
o_shippriority 0
o_comment jkACLh 0igMiy72n Sky0h0B6NB70j7Q

o_orderkey 34
o_custkey 3744836

o_orderstatus O
o_totalprice 112299.56
o_orderdate 1998-07-21
o_orderpriority 3-MEDIUM
o_clerk Clerk#000006684
o_shippriority 0
o_comment 05k 2x242klm jyA
wB0CBzzQnz5P11nAml5AL5jC lg5

10 row(s) retrieved.

G-3: Part contents

p_partkey 1
p_name goldenrod lace spring chartreuse ivory
p_mfgr Manufacturer#1
p_brand Brand#13
p_type PROMO BURNISHED COPPER
p_size 7
p_container JUMBO PKG
p_retailprice 901.00
p_comment zMg1PACmQ 7RCCC7

p_partkey 2
p_name snow ghost azure burnished lemon
p_mfgr Manufacturer#1
p_brand Brand#13
p_type LARGE BRUSHED BRASS
p_size 1
p_container LG CASE
p_retailprice 902.00
p_comment Bxg4RlO6051n7NjN zn

p_partkey 3
p_name cornflower navajo salmon lemon orchid
p_mfgr Manufacturer#4
p_brand Brand#42
p_type STANDARD POLISHED BRASS
p_size 21
p_container WRAP CASE
p_retailprice 903.00
p_comment 4241RR3By

p_partkey 4
p_name olive dim lemon light khaki
p_mfgr Manufacturer#3
p_brand Brand#34
p_type SMALL PLATED BRASS
p_size 14
p_container MED DRUM
p_retailprice 904.00
p_comment z1n7znz6

p_partkey 5
p_name lavender cornsilk linen seashell lemon
p_mfgr Manufacturer#3
p_brand Brand#32
p_type STANDARD POLISHED TIN
p_size 15
p_container SM PKG
p_retailprice 905.00
p_comment gj4Lg5BhBk12iS

p_partkey 6
p_name cornsilk beige chartreuse medium blue
p_mfgr Manufacturer#2
p_brand Brand#24
p_type PROMO PLATED STEEL
p_size 4
p_container MED BAG
p_retailprice 906.00
p_comment yNjzS Njyh4mgLx Om

p_partkey 7
p_name honeydew purple cream mint coral
p_mfgr Manufacturer#1
p_brand Brand#11
p_type SMALL PLATED COPPER
p_size 45
p_container SM BAG
p_retailprice 907.00
p_comment PSNg0L

p_partkey 8
p_name puff blush tomato papaya navy
p_mfgr Manufacturer#4
p_brand Brand#44
p_type PROMO BURNISHED TIN
p_size 41
p_container LG DRUM
p_retailprice 908.00
p_comment k042AL4y21N1yNPC77

p_partkey 9
p_name burnished violet pink rose drab
p_mfgr Manufacturer#4
p_brand Brand#43
p_type SMALL BURNISHED STEEL
p_size 12
p_container WRAP CASE
p_retailprice 909.00
p_comment 37PLkwhgiAP0xCkxO

p_partkey 10
p_name slate dark white lavender purple
p_mfgr Manufacturer#5
p_brand Brand#54
p_type LARGE BURNISHED STEEL
p_size 44
p_container LG CAN
p_retailprice 910.01
p_comment wPP74M1Lwj1

10 row(s) retrieved.

G-4: Partsupp contents

ps_partkey 1
ps_suppkey 2
ps_availqty 3325
ps_supplycost 771.64
ps_comment
00PL56QkQRSkg2z7MANNj4i1h2zLQQLiQnAlML1S6
k4hg3hP5hk3ywMLwy 7gjR
3 4Q7S1Qmzx2jOS37Mk61n
yCg4Q7k522P0055wg23B0Mw3BOwSy6z5Q6xljABx3
LAj6R6CmM
l4jjMzQ02LkiiyCCwBk7w465kLBz7QlCk26ARLOxk7z2hC
0jw7

ps_partkey 1
ps_suppkey 75002
ps_availqty 8076
ps_supplycost 993.49
ps_comment
nSO7Mln4N7LlxgAyM2MzNnO7k0NlhjyShgCy30A
27QML0SQ77CPPgkCQAQCwz5M
3MmSSAQ LxMLMC0Bj4CNN276SmQRSljPxz5z3
L2mLMQSBghjLnCOR4Nl 156OMP
C76QL

xiyw0kSQy1w6ygAxOA4hx7Ngihi5NAPNl2LQ4SRnNhn7m
ygOB0z

ps_partkey 1
ps_suppkey 225002
ps_availqty 4069
ps_supplycost 357.84
ps_comment 234OCA5ghw0P0gS3n2jCS35yAm 3L5C7iB
k7 w1 R52LLOACQ6i606OB 2MP1x0

wC23ik2Omk4NnxzmS6z5z5il661l2l2g0P3OLk66jzQxjSSA
nwSnQ3xz QBQR2j1
0hNmmyQ14hl45l4x5C B5Qz
Lk26yhQNmS54A2O7wlC P0C57CyxSL3

ps_partkey 1
ps_suppkey 150002
ps_availqty 3956
ps_supplycost 337.09
ps_comment 6215k
jLCizNlOBl62nP4lLQy431kOzyzn2M6L3h73lCljhlx3x5ghjl
OyL76A0h
zPk2CS2jkxN gAN3gnk652 Cj4k4

ps_partkey 2
ps_suppkey 3
ps_availqty 8895
ps_supplycost 378.49
ps_comment MMNOM3BnMM6NBzjB 2mg i jALB
nQhBM5ROi5N7A5w4B4S2k1506OzMgh6SRB7n

P1hQCjgjR17SBA77g6niCwi0L6Pghh1S0O4mlSL1OShRky
xQS7NNQj

ps_partkey 2
ps_suppkey 75003
ps_availqty 4969
ps_supplycost 915.27
ps_comment
6S66zNlykhii26wwAxz1PRMxggAy446yylPBS5wP
Rl6ggNkyikkyhxymMShNgQm
Bim1N60
00NSjwPw02lhPPSmn3yRSCn105llnPBk2MlRlxxQAmR
mO2kxiRh5Pk4
x2OnS40nnQRm
16L6NC2RSAkL36g6w64L5w4w74Lnz5wROSOQNingx4i
mSnPwz5N
hSxNg

ps_partkey 2
ps_suppkey 225003
ps_availqty 3025
ps_supplycost 306.39
ps_comment
y5BNy3Aw02nxyMxgzP5BS14gg7MCnLlSkih56gOMOy4Q
yNj5P3iM jOmkRQhRR3h
ylCn4jN LlgSxyPigjRLgBygM
RR4CL3Pjx6CRMNl1iA7w2ALwkzn06khOOzyQNB
63wP4BLSk wPNk4My

ps_partkey 2
ps_suppkey 150003
ps_availqty 8539
ps_supplycost 438.37
ps_comment BPOgj3k MgQR2
x6kn3BR6lkmhMzjQk6S343LmN6lz2Qh512MSi4nQ5Bghxl
h401

yyn463mxOh7Bxgx1PR60jwSMQ661OCA6mPO33R05R2S
6N330in0Qx0AC4QhBnOkz
4NlQkMR2gh3 k1xk SjN40C77hm2Qlh01jAkSP

ps_partkey 3
ps_suppkey 4
ps_availqty 4651
ps_supplycost 920.92
ps_comment P7 437MmnM0Pik
lAwBj0gSnm1z1zAMA6417zgS154nLCC0Q6BC1lgxyB6Bk
Oj6Q
CC6n4mw2w7jgCNP zz5AMw37 z

ps_partkey 3
ps_suppkey 75004
ps_availqty 4093
ps_supplycost 498.13
ps_comment
PyRmlwO76kO3igxhS64h5x6PBLM2PxkO0j3NMRgzP6S1g
hhw4Nnn04my03lzCyQQ
4M3gS2kO2iOmxPzlCM330zN1yPnzml1ixgB

10 row(s) retrieved.

G-5: Customer contents

c_custkey 1
c_name Customer#000000001
c_address ANhzAAh6R3 glS4Sx
c_nationkey 15
c_phone 25-989-741-2988
c_acctbal 711.56
c_mktsegment BUILDING
c_comment
j5S37kk6zkOzkM5NOz6jwwimkN66CmOhM5ySy
w6PAj2xjOAmhkw6ChSR 21BMRkL
0kLM5zxg654CRlB3 1Lxm3S

c_custkey 2
c_name Customer#000000002
c_address MN0L3OzNgylx2
c_nationkey 13
c_phone 23-768-687-3665
c_acctbal 121.65
c_mktsegment AUTOMOBILE
c_comment M4QB23ixkg0yk6m3gwim6zi32PS7lj2

c_custkey 3
c_name Customer#000000003
c_address PSL74SNCwwN2ON66lxgnw7mR4hLP2k
c_nationkey 1
c_phone 11-719-748-3364
c_acctbal 7498.12
c_mktsegment AUTOMOBILE
c_comment mSC13MBj4n0P6Mgh0ml02zOBlyjw3NzB1

c_custkey 4
c_name Customer#000000004
c_address mknn1Sh0NPMz1k5Lw2OB mO
c_nationkey 4
c_phone 14-128-190-5944
c_acctbal 2866.83
c_mktsegment MACHINERY
c_comment
MN6ChhSMwPwzOkyww7C5ROlhMS0C4iR2nC6kQmywx3
yim62QNySOmQRQnwizihMOg

c_custkey 5
c_name Customer#000000005
c_address yOww5znhPNi5OlQNPChkLx2BLPxNSB
c_nationkey 3
c_phone 13-750-942-6364
c_acctbal 794.47

c_mktsegment HOUSEHOLD
c_comment 24BOSzg 03m710wll
iNxnwQ0OmzzgO7A3ykBj2 g755hhCyMO7QnARx5Pg3kyA
QA35
i0CS1MSLg0xN2iyg0liwnMwnOx52nj5iQkNQPP

c_custkey 6
c_name Customer#000000006
c_address nS70ykL4n k51ik3R5wlNzjnjBL2N51ki
c_nationkey 20
c_phone 30-114-968-4951
c_acctbal 7638.57
c_mktsegment AUTOMOBILE
c_comment hPMLmxPw05R1mz126jjRAj1kOP7xLC6
yS3ALCRBR5B3im650BLm4O3SwBP7xlwOk
lmPRS31RNN0gMkkPm4COigCRMlniz27jwg63yz

c_custkey 7
c_name Customer#000000007
c_address ChljB04OgAizN6kQhRi7LjjNiCM0A AS
c_nationkey 18
c_phone 28-190-982-9759
c_acctbal 9561.95
c_mktsegment AUTOMOBILE
c_comment QM63L2miSw3hy34iQ11235
011mkgk0SkCRC73L1CgiLROzNwjO4PQSBx2n2iQg5h

c_custkey 8
c_name Customer#000000008
c_address kCRz0CknMw7mh4P50QjBnxSLRxQCMOAh
yNn
c_nationkey 17
c_phone 27-147-574-9335
c_acctbal 6819.74
c_mktsegment BUILDING
c_comment x1Rh1P5M73Lix xyM
Lmng0RO4MBQyLll7wzwgyOLCxi2yCLgL1zO4yOiAPj

c_custkey 9
c_name Customer#000000009
c_address L4z65g2RNgN6PxM5kRjnPB7k2kwL62
c_nationkey 8
c_phone 18-338-906-3675
c_acctbal 8324.07
c_mktsegment FURNITURE
c_comment
7zRiSzmj4Ak7L6N7RljhM5437B6CPmP54RC1x1x7C6hziN
6l

c_custkey 10
c_name Customer#000000010
c_address L3jg3xAwi6A0B103B0Aymm
c_nationkey 5
c_phone 15-741-346-9870
c_acctbal 2753.54
c_mktsegment HOUSEHOLD
c_comment 7Lm
LiCwwxQMykgNOR6kzCyP1B21QyA57hBlSOPnx6m53iSO
P6w44M3CP MnP7Alk
y4OwkOwSh20341

10 row(s) retrieved.

G-6: Supplier contents

s_suppkey 1
s_name Supplier#000000001
s_address N kw4gn1OM Ahw3Sg70BBgQw57lgjzj55R
s_nationkey 17
s_phone 27-918-335-1736
s_acctbal 5755.94
s_comment lLniMi51QPmO1 C2hy27wkN21mmg53
BhQBB1O2x4OmiR4kO5kN1BS 4PwMhk Pk2n
RnA2 k

s_suppkey 2
s_name Supplier#000000002
s_address ji3yh016B5
s_nationkey 5
s_phone 15-679-861-2259
s_acctbal 4032.68
s_comment B32z0yzh2lPyOwQkAjA704yM2R7llRlk2
xCliy41QNmQn0RNl0OQ4jgMy3kSRBLzy
w25CB5 lk0A 54

s_suppkey 3
s_name Supplier#000000003
s_address mxBQBnxO3CSwl7
s_nationkey 1
s_phone 11-383-516-1199
s_acctbal 4192.40
s_comment
BS0OzjiO1yM6Rgl4mxLNhjSMPB37Sw7ym3R7ll2n4SSCilz
6nlL5SBOig

s_suppkey 4
s_name Supplier#000000004
s_address 7zR323R73NMB77wi1
s_nationkey 15
s_phone 25-843-787-7479
s_acctbal 4641.08
s_comment w
lQn6QyOSSxhw10C6gz2BngiLRAMmgnRxiLiO3

s_suppkey 5
s_name Supplier#000000005
s_address AmMQ7Mg
10ByLCP52M13xN31jh5hzOgnm00B
s_nationkey 11
s_phone 21-151-690-3663
s_acctbal -283.84
s_comment PAziBQQixjwS7P4lQhn10i74050M
AzkxACnOAyjnSm3CQ26SOx5kynSR0nlLSzi3y
3nzPPNikNl3P3 kLwwOP7AM30CO0ymAh

s_suppkey 6
s_name Supplier#000000006
s_address QQL6hxmnMkkgMwgm7CB5B 30Llz
s_nationkey 14
s_phone 24-696-997-4969
s_acctbal 1365.79
s_comment giSki24 gRNAmB
1yOzPR6Q2kiNCQ0h3LLyxmROA5O7OOi5zlzy

s_suppkey 7
s_name Supplier#000000007
s_address z45m2jBRzl5ilLNz4
s_nationkey 23
s_phone 33-990-965-2201
s_acctbal 6820.35
s_comment 1PhngjmiSQl0RzRACP0l4S70xSL
QPSBM16072SkMLCgm4OOMjARLNQk3g1P3BB32A
gBMl462B0CP7Rh24

s_suppkey 8
s_name Supplier#000000008
s_address xz5m4C A4AAj0kANQ
s_nationkey 17
s_phone 27-498-742-3860
s_acctbal 7627.85
s_comment lz57Mw6RNwCSCzmAShwN7S45w20C5zS6zi

5Al1ORMwnQmjS5SgBnRhQ1lCkyBlhN
6MP7 kAzNw3gSjyyLMiNzhCmPn0 g5x23Q

s_suppkey 9
s_name Supplier#000000009
s_address m7k7CnC3wiP
s_nationkey 10
s_phone 20-403-398-8662
s_acctbal 5302.37
s_comment xPLzNgk5nzA jm3PLmySlm PS
zANRjSgh2njAg

s_suppkey 10
s_name Supplier#000000010
s_address wN1S4mQ0g7Px5Lj34xw6kS4Li4NzB4mO
s_nationkey 24
s_phone 34-852-489-8585
s_acctbal 3891.91
s_comment
5xwg6AOz0NzhONL6kC43zR3AhzO6njCiwPg7k6MxwP1m
N2 Rg 5Q426

10 row(s) retrieved.

G-7: Nation contents

n_nationkey 0
n_name ALGERIA
n_regionkey 0
n_comment 2Cxhl7
L1iwk6hMh300izngN32CPwCikyLk6khMzSRA

n_nationkey 1
n_name ARGENTINA
n_regionkey 1
n_comment
zQn3Okwz1wLn7PLS3OhCgn56kP5PyRikgi1B7lL

n_nationkey 2
n_name BRAZIL
n_regionkey 1
n_comment gLmS0nACAmnBCj2klki7RCPNgPxnCOjNg4k
OiAg57COSOm1NwCnOyLx40R SC y20
gPPAkNk5hxRhR5OmgS1iPQQzNAxPL30n67OgyC
l617Sh4LS

n_nationkey 3
n_name CANADA
n_regionkey 1
n_comment 4yMO AhnQ5Lh
wzQAM662Aw1ByCl7CxmzRwNR5nAlO4 x

n_nationkey 4
n_name EGYPT
n_regionkey 4
n_comment l1im5126 Cxj
NMQmLxOikni02j2m3Ah4yNR1QQiL507j2QSlyN

n_nationkey 5
n_name ETHIOPIA
n_regionkey 0
n_comment NS7n LSOP Oz5n1AlB2S02nN0lMh4SBxP
iRhBO 047R26 2BlM

n_nationkey 6
n_name FRANCE
n_regionkey 3
n_comment 3mjmizl S 3L3k2hNNhNlP4w370xRxyN15wn

n_nationkey 7
n_name GERMANY
n_regionkey 3
n_comment z nOP4RkwO CmzBB 516mAg
lByw4OM3QyNPA

n_nationkey 8
n_name INDIA
n_regionkey 2
n_comment MNlR5RCiRMj1l11wjN7Myn
M1lylNlMmBQl7PL4C kkxQkgPQ7i3w6B67R2QkOO40
xl4Q2iw76jRL7ilhR5Q
0xC7RRm5iQ2NAx2LiBm3QiO27j

n_nationkey 9
n_name INDONESIA
n_regionkey 2
n_comment SjPmQO71Lj
7ABj6MxlAQk3nLwi73BPxzCwjzMn4zlLzgg6nnz0j0w
zxC66gP6ykR
PMg

10 row(s) retrieved.

G-8: Region contents

r_regionkey 0
r_name AFRICA
r_comment
xSx31zz31Cl1z4OAnmm05AjiOxC3AMMNOgC0kACgwngg
3glP7LLLywlQy7R

r_regionkey 1
r_name AMERICA
r_comment
kgyh3LSnC72k6zlAz0LP3k2L4QB1QL1O673OjO1SPj0ngQ
7CO100SBgmgRQ4lgPCMk
21A425iklyAR4yBRAwR4Cm5miNw
4jl13mMnxw17B

r_regionkey 2
r_name ASIA
r_comment NSg6xlMlA1lzm6mOR0Ajx
nhRA77NgRxBwL1M6Py RjySB3RLwkyPkwMM2R1BQ
xAz
kOgkjmll0gAghinP5inmNmR76MlijMS3S2zxONR15

r_regionkey 3
r_name EUROPE
r_comment
zlSL7Qwg12hMBL5lhlz0M45QkjShwSyiO04MLOh7wn1AR
LQPyPAyAiil576l1Li7Al
nR1S RQ4SLny7B2Ryj5P66MLhn NxhwB4C3ig0SO

r_regionkey 4
r_name MIDDLE EAST
r_comment RllxmhPLz3Cy2mNlg4QMBnNASM ACki
MPki7Oi

5 row(s) retrieved.