UNISYS

TPC Benchmark™ H Full Disclosure Report

Unisys ES7000 Aries 420 Enterprise Server

using

Oracle Database 10g Enterprise Edition

on

Microsoft Windows Server 2003, Datacenter Edition for 64-bit Itanium-based Systems

August 13, 2004

Unisys Part Number: QDOC # 1000

First Printing - August 13, 2004

Unisys believes that the information in this document is accurate as of the publication date. The information in this document is subject to change without notice. Unisys Corporation assumes no responsibility for any errors that may appear in this document.

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

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

All performance data contained in this report was obtained in a rigorously controlled environment, and therefore results obtained in other operating environments may vary significantly. Unisys Corporation and Microsoft Corporation do not warrant or represent that a user can or will achieve similar performance expressed in composite query-per-hour ratings. No warranty of system performance or price/performance is expressed or implied with this document.

Unisys assumes no responsibility for any errors that may appear in this document. Unisys reserves the right to make changes in specifications and other information contained in this document without prior notice, and the reader should in all cases consult Unisys to determine whether any such changes have been made.

Printed in USA, August 2004

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

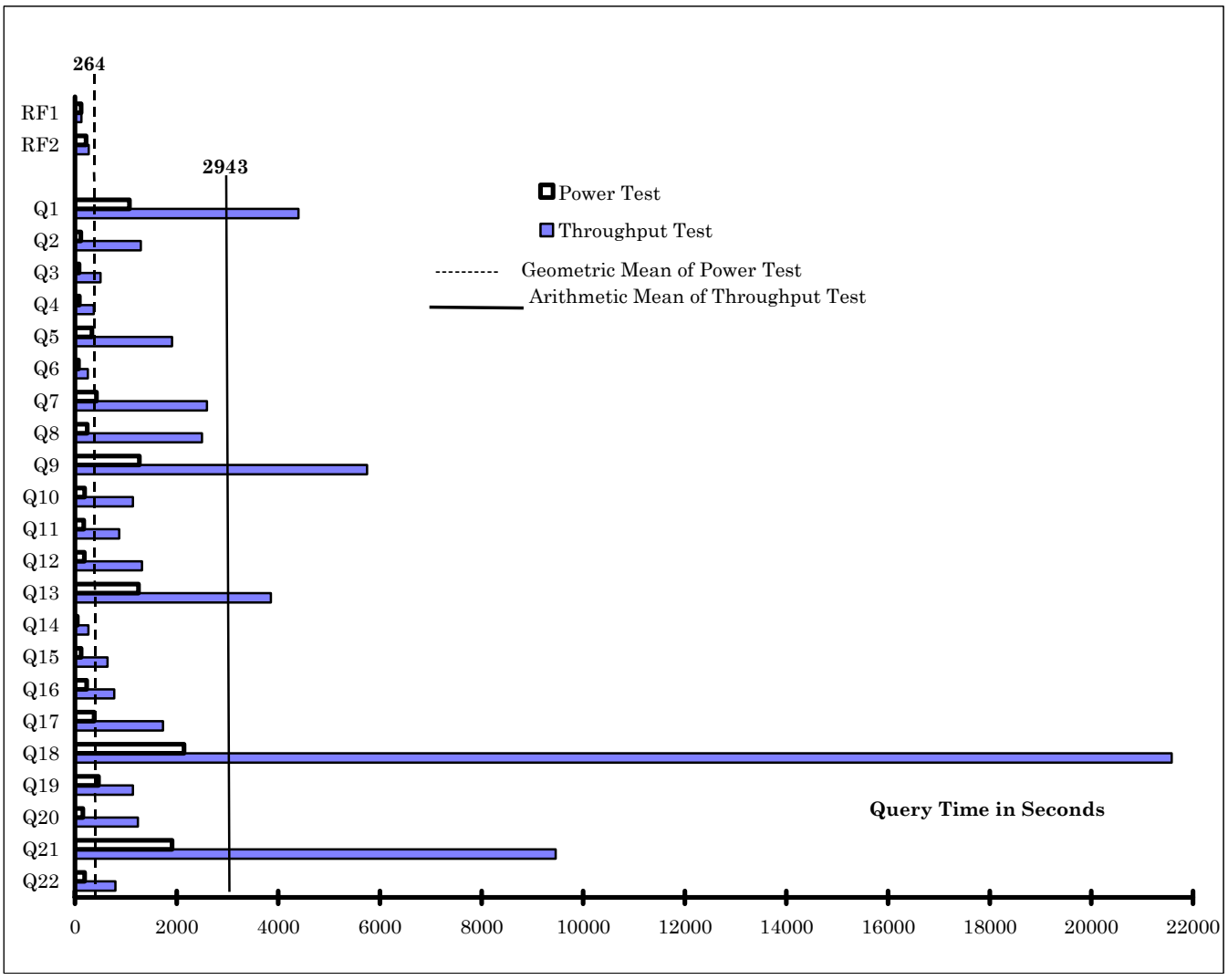
TPC Benchmark™	Trademark of the Transaction Processing Performance Council
TPC-H, QppH QthH, and QphH	Trademark of the Transaction Processing Performance Council
Microsoft	Trademark of the Microsoft Corporation
Oracle Database 10g	Trademark of the Oracle Corporation
Windows Server 2003	Trademark of the Microsoft Corporation
Unisys	Trademark of the Unisys Corporation

Other product names used in this document may be trademarks and/or registered trademarks of their respective companies.

UNISYS	ES7000 Aries 420 Enterprise Server	TPC-H Rev. 2.1.0 Report Date August 13, 2004

Total System Cost	Composite Query per Hour Rating	Price Performance
$669,239	9,853.3 QphH @ 1000GB	$68 / QphH @ 1000GB

Database size	Database Manager	Operating System	Other Software	Availability Date
1000 GB	Oracle Database 10g Enterprise Edition 64-bit with Automatic Storage Management	Microsoft Windows Server 2003, Datacenter Edition for 64-bit Itanium-based Systems	MKS Toolkit for Developers 8.6	November 30, 2004 *



Database Load Time = 5:25:43 **Load included backup: N** **Total Data Storage / Database Size = 5.67**

RAID (Base tables): Y **RAID (Base tables and Auxiliary Data Structures): Y** **RAID (All): N**

System Configuration		
Processors	16 x 1.5 GHz Intel® Itanium2™ with 6MB Level 3 Cache	
Memory	64 GB Main Memory	
Disk Controllers	26	PCI Fibre Channel
Disk Drives	164	36GB FC (34.17 GB useable)
	2	36GB SCSI (33.47 GB useable)
Total Disk Storage	5670.82 GB	

* All hardware is available now

UNISYS

ES7000 Aries 420 Enterprise Server

TPC-H Rev. 2.1.0

Report Date
August 13, 2004

	Part No.	Third Party Brand	Third Party Price	Unit Price	Qty.	Extended Price	3 yr.Maint. Price
SVR: ES7000-420, 16x 1.5GHz/6MB, 32GB Mem.	ES7420166-100		1	$260,300	1	$260,300	$12,936
1x RAID Controller, 2x 36GB Disk (boot media)	Included						
Sentinel System Management S/W and Media	Included						
36Ux19"x34" cabinet and skins	Included						
MEM: 32GB, 1 GB DIMMs	MEM321-32G		1	$22,800	1	$22,800	
I/F:4-Slot I/O Mod, 100M	MOD3000-PCI		1	$1,805	4	$7,220	
CTRL: Ethernet, 1000Mb, 1-Port Opt., PCI	ETH33311-PCX		1	$499	1	$499	
I/F: Monitor, 15-inch Color, Kybrd, Mse & Cable	ES70003-UIF		1	$570	1	$570	
				Server Subtotal		**$291,389**	**$12,936**
CTRL: PCIX, Fiber Channel, 2Gb, 1-Port	FCH742313-P64		1	$1,330	26	$34,580	
DISK: 3.0TB Pkg, 84x 36GB, 15Krpm, + 10% spares *	JBD203615-P84		1	$36,086	2	$72,172	Spared
DISK: 0.5 TB Pkg, 14x 36GB, 15Krpm, + 10% spares *	JBD203615-P14		1	$8,772	1	$8,772	Spared
RCK: FC 2Gb Storage, Optical/Optical I/F	JBD2020-SA		1	$3,958	13	$51,454	$8,112
RCK: Rack Mount Kit, Disk Enclosure	JBD2001-RMK		1	$190	13	$2,470	
CBL: FC, Optical, LC Conn's, 10m + 10% Spares *	GCFAZLL10	3	3	$36	26	$936	Spared
PWR: Distribution Strip, 9-Plug w/ Line Cord	SFR91-PWR		1	$238	3	$714	
CAB: 36U x 19" x 41" Open Front Cabinet	HRT361941-OFT		1	$1,425	1	$1,425	
				Storage Subtotal		**$172,523**	**$8,112**
O/S: Windows 2003 Svr, DataCenter Ed, 16P, 1yr Lsub.	WND2316-TSP		1	$35,112	1	$35,112	
O/S: Windows 2003 Server, DC Lmtd Sbscrptn, 1yr.	DUS200316-TSP		1	$3,173	2	$6,346	
O/S: Windows 2003 Server DC Support, 24x7x4, 1yr.	GTS7412316-SSA		1	$12,574	3		$37,722
Oracle Database 10g Enterprise Edtn, Named User Plus for 3 yrs		4	4	$10,000	16	$160,000	Inc. below
Oracle Partitioning, Named User Plus for 3 years		4	4	$2,500	16	$40,000	Inc. below
Oracle Database Server Support Package for 3 years		4	4	$6,000	1		$6,000
MKS Toolkit for Developers		2	2	$479	1	$479	Inc. above
				Software Subtotal		**$241,937**	**$43,722**
				Total		**$705,849**	**$64,770**
Oracle Mandatory E-Business Discount (license and support)						($30,900)	
Computer Resolutions Inc. 13% Large Volume Discount						($70,480)	

Notes:
1. * = 10% or minimum 2 spares added in place of onsite service.
2. HW & SW maintenance at 24 x 7 w/ 4 hr. max. response time for spares.
3. 3rd Party Brand & Pricing: 1 = CRI Product Price, Unisys Maintenance Price
   2 = Microsoft 3 = GoCables 4 = Oracle; pricing contact: MaryBeth Pierantoni, mary.beth.pierantoni@oracle.com, 650-506-2118

Three Year Cost of Ownership: $669,239
QphH @ 1TB: 9,853.3
$ / QphH@1TB: $67.93

Benchmark results and test methodology audited by Lorna Livingtee of Performance Metrics, Inc.

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 assumption 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 benchmarks specifications. If you find that the stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank You.

Numerical Quantities Summary

Measurement Results

Scale Factor	1000 GB
Total Data Storage / Database Size	5.67
Start of Database Load	7/17/2004 10:40:37
End of Database Load	7/17/2004 16:05:20
Database Load Time	5:24:43
Query Streams for Throughput Test	7
TPC-H Power	13614.2
TPC-H Throughput	7131.3
Composite Query per Hour Rating (QphH@1000GB)	9853.3
Total System Price Over 3 Years	$669,239
TPC-H Price Performance Metric	$67.93

Measurement Intervals

Measurement Interval in Throughput Test (Ts)	77742	seconds

Duration of Stream Execution:

	Seed	Query Start Date/Time Query End Date/Time	RF1 Start Date/Time RF1 End Date/Time	RF2 Start Date/Time RF2 End Date/Time	Duration
Stream 0	717160520	7/18/04 20:56:33 7/19/04 0:01:42	7/18/2004 20:54:30 7/18/2004 20:56:32	7/19/04 0:01:42 7/19/04 0:05:21	3:10:51
Stream 1	717160521	7/19/04 0:05:40 7/19/04 16:52:32	7/19/04 20:54:58 7/19/04 20:56:58	7/19/04 20:56:58 7/19/04 21:00:38	20:54:58
Stream 2	717160522	7/19/04 0:05:40 7/19/04 18:51:11	7/19/04 21:00:38 7/19/04 21:02:45	7/19/04 21:02:45 7/19/04 21:08:21	21:02:41
Stream 3	717160523	7/19/04 0:05:41 7/19/04 18:36:27	7/19/04 21:08:21 7/19/04 21:10:32	7/19/04 21:10:32 7/19/04 21:14:23	21:08:42
Stream 4	717160524	7/19/04 0:05:41 7/19/04 18:13:52	7/19/04 21:14:02 7/19/04 21:16:30	7/19/04 21:16:30 7/19/04 21:20:01	21:14:20
Stream 5	717160525	7/19/04 0:05:41 7/19/04 14:29:14	7/19/04 21:20:14 7/19/04 21:22:24	7/19/04 21:22:24 7/19/04 21:27:50	21:22:09
Stream 6	717160526	7/19/04 0:05:41 7/19/04 17:48:04	7/19/04 21:27:50 7/19/04 21:30:03	7/19/04 21:30:03 7/19/04 21:33:46	21:28:05
Stream 7	717160527	7/19/04 0:05:41 7/19/04 20:54:45	7/19/04 21:33:46 7/19/04 21:35:58	7/19/04 21:35:58 7/19/04 21:41:15	21:35:34

TPC-H Timing Intervals (in seconds):

Query	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Stream 0	1067.9	117.5	79.2	84.0	334.4	67.0	420.7	242.3
Stream 1	4579.8	1595.1	591.6	502.6	2254.8	108.6	3410.0	1238.8
Stream 2	5455.5	1540.0	480.0	359.6	2155.6	1032.7	3234.4	1112.0
Stream 3	4804.2	1733.3	168.1	562.3	2392.2	166.7	3569.9	8689.4
Stream 4	5192.1	1347.1	242.7	425.1	2284.0	139.0	3149.4	863.7
Stream 5	4372.0	1484.7	327.4	165.2	2395.9	135.7	1130.1	1375.7
Stream 6	5286.4	1246.6	1609.2	467.5	1533.9	120.9	3241.5	1547.6
Stream 7	1064.2	110.3	87.8	83.0	347.3	64.1	424.3	2646.5
Min Qi	1064.2	110.3	87.8	83.0	347.3	64.1	424.3	863.7
Max Qi	5455.5	1733.3	1609.2	562.3	2395.9	1032.7	3569.9	8689.4
Avg Qi	4393.5	1293.9	501.0	366.5	1909.1	252.5	2594.2	2496.2
Query	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16
Stream 0	1268.3	189.6	172.8	182.8	1250.9	48.5	122.9	227.9
Stream 1	6075.4	1257.8	1021.4	985.7	4319.1	362.5	622.2	1142.3
Stream 2	8217.5	1698.4	1294.6	1394.1	2780.5	203.0	530.3	1092.1
Stream 3	7158.1	804.9	841.0	310.3	3515.8	230.3	757.6	458.6
Stream 4	7167.2	1243.8	973.9	1599.8	5384.1	278.3	400.6	732.3
Stream 5	5094.1	895.4	942.2	2601.1	3983.6	432.4	183.9	538.8
Stream 6	5455.0	1883.8	816.8	2040.3	5761.3	281.7	1860.3	1199.0
Stream 7	1065.5	172.0	174.5	272.8	1251.8	47.9	121.7	231.3
Min Qi	1065.5	172.0	174.5	272.8	1251.8	47.9	121.7	231.3
Max Qi	8217.5	1883.8	1294.6	2601.1	5761.3	432.4	1860.3	1199.0
Avg Qi	5747.5	1136.6	866.3	1314.9	3856.6	262.3	639.5	770.6
Query	Q17	Q18	Q19	Q20	Q21	Q22	RF1	RF2
Stream 0	372.7	2146.1	460.1	156.3	1908.1	189.2	122.5	218.5
Stream 1	929.1	19441.8	1027.6	2927.9	5193.7	824.0	120.0	220.1
Stream 2	7314.4	14830.0	1342.9	881.7	9368.2	1212.6	126.7	335.8
Stream 3	952.7	15024.5	1461.4	921.1	11375.4	748.7	131.3	230.7
Stream 4	733.8	14056.8	985.9	429.8	17198.2	463.6	127.1	224.1
Stream 5	740.4	16434.2	1045.5	1155.5	5235.5	1144.4	129.5	326.4
Stream 6	1077.8	14981.6	1650.6	917.6	9805.6	958.0	132.3	223.0
Stream 7	384.9	56289.5	464.7	1415.3	8035.8	189.4	132.1	316.6
Min Qi	384.9	14056.8	464.7	429.8	5193.7	189.4	120.0	220.1
Max Qi	7314.4	56289.5	1650.6	2927.9	17198.2	1212.6	132.3	335.8
Avg Qi	1733.3	21579.8	1139.8	1235.6	9458.9	791.5	128.4	268.1

PERFORMANCE METRICS INC.
TPC Certified Auditors

August 11, 2004

Jerrold Buggert
Director of Modeling and Measurement
Unisys Corporation
25725 Jeronimo Road
Mission Viejo, CA 92691

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

Platform: ES7000 Aries 420
Database Manager: Oracle Database 10g Enterprise Edition
Operating System: Microsoft Windows 2003 Server Datacenter Edition

CPU's	Memory	Total Disks	Qpph@ 1000GB	QthH@1000GB	QphH@1000GB
16 Intel Itanium MP @ 1.5 Ghz	64 GB	166 @ 36GB	**13,614.2**	**7,131.3**	**9,853.3**

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

- The database tables were defined with the proper columns, layout and sizes.
- The tested database was correctly scaled and populated for 1,000GB using DBGEN. The version of DBGEN was 1.3.0.
- The qualification database layout was identical to the tested database except for the number and size of the files.
- The query text was verified to use only compliant variants and minor modifications.
- The executable query text was generated by QGEN and submitted through Oracle's standard interactive interface. The version of QGEN was 1.3.0.
- The validation of the query text against the qualification database produced compliant results.

- The refresh functions were properly implemented and executed the correct number of inserts and deletes.
- The load timing was properly measured and reported.
- The execution times were correctly measured and reported.
- The performance metrics were correctly computed and reported.
- The repeatability of the measurement was verified.
- The ACID properties were tested and verified.
- Sufficient mirrored log space was present on the tested system.
- The system pricing was checked for major components and maintenance.
- The executive summary pages of the FDR were verified for accuracy.

Auditor's Notes:
Special attention was given to the query text of Query 15. This query used an approved variant (A) and a minor query modification allowed for select list expression aliases. This query was compliant with both requirements.

Sincerely,

Lorna Livingtree
Auditor

PO Box 984 Klamath, CA 95548
(707) 482-0523 fax: (707) 482-0575
email: LornaL@PerfMetrics.com

From: MaryBeth Pierantoni [mailto:mary.beth.pierantoni@oracle.com]
Sent: Tuesday, June 29, 2004 6:59 PM
To: Mutchler, David G
Cc: Williams, Graham T.
Subject: Re: TPC-H pricing
Hi David,

To follow is pricing information you requested. I understand Partitioning is being used, so I included this option in the pricing.

Product	Price	Quantity	Extended Price
Oracle Database 10g Enterprise Edition, Named User Plus for 3 years	10,000	16	160,000
Partitioning, Named User Plus for 3 years	2,500	16	40,000
Oracle Database Server Support Package for 3 years	6,000	1	6,000
Oracle Mandatory E-Business Discount			<$30,900>
TOTAL			$175,100

Contact: MaryBeth Pierantoni, mary.beth.pierantoni@oracle.com, 916-315-5081

Please let me know if you have any questions.

Thanks,
MaryBeth

GoCables.com - Fiber optics, fiber optic, fiber optic cable, fiber optic patch cables, fiber cables, s... Page 1 of 1

LC to LC - Multimode Duplex Fiber Optic Patch Cable. LC cables are avaliable in 50µ and 62.5µ (62.5 is Standard) and in *any cable length.*

The LC fiber optic connector interface is based on a 1.25mm ceramic ferrule and the familiar latching mechanism of the RJ-45 modular plug and jack. The LC connector is a small-form-factor connector that offers a port density equal to the modular jack.

Standard Length Cables						Add to Cart
Part #	Cable Length	Unit Price	5-9 20% Off	10-19 25% Off	20 or more 30% Off	
GCFAZLL01	1 m	34.08	27.26	25.56	23.86	
GCFAZLL02	2 m	35.63	28.50	26.72	24.94	
GCFAZLL03	3 m	37.18	29.74	27.89	26.03	
GCFAZLL05	5 m	40.28	32.22	30.21	28.20	
GCFAZLL10	10 m	48.03	38.42	36.02	33.62	
Custom Length Cable Pricing						Click Here

http://gocables.com/Fiber_Optic_Patch_Cables/index.htm 7/12/2004

COMPUTER® RESOLUTIONS
Taking IT Personally℠
Phone: 800-848-7185
Fax: 203-384-0473
35 Benham Avenue
Bridgeport, CT 06605
www.cri1.com

SALES QUOTE

SQ-17925	Jul 12,2004

Customer	Contact	Ship To
UNISYS CORPORATION DAVID MUTCHLER 2515 TERMINAL ROAD DOCK F ROSEVILLE MN 55113 UNITED STATES Tel :(651) 635-5759 Fax :(651) 635-7523		UNISYS CORPORATION DAVID MUTCHLER 2515 TERMINAL ROAD DOCK F ROSEVILLE MN 55113 UNITED STATES Tel :(651) 635-5759 Fax :(651) 635-7523

Account	Terms	Due Date	Account Rep	Schedule Date
UNISYS-Q	NET 30	Aug 11, 2004	John Palmieri	

Quotation	PO #	Reference	Ship VIA	Page	Printed
SQ-17925			UPS GROUND	1	08/10/04 11:17

L	Item	Description	Qty	Ship	Price	M	Discount	Amount
1		ES7000-420 SERVER						
2	ES7420166-100	SVR:ES7/420 1.5GHZ 1GB	1		260300.00	EA		260300.00
3	MEM321-32G	ME:32GB MEMORY PKG	1		22800.00	EA		22800.00
4	MOD3000-PCI	I/F:4-SLOT I/O MOD, 100M	4		1805.00	EA		7220.00
5	ETH33311-PCX	LAN HW:GBIT ETH OPT	1		499.00	EA		499.00
6	ES70003-UIF	RCK:105 KBRD/MOUSE/MON	1		570.00	EA		570.00
7	WND2316-TSP	O/S:WINDOWS 2003 SERVER, DC, 16P,1YR LSU	1		35112.00	EA		35112.00
8	DUS200316-TSP	O/S:DC 16X LTD. SUB 1 YR	2		3173.00	EA		6346.00
9	GTS7412316-SSA	ES7K410-30/16 SW SPT+SSA	3		12574.00	EA		37722.00
10		STORAGE HARDWARE						
11	FCH742313-P64	CTRL: FIBRE CHAN 2 66 LP9802 2GH HBA	26		1330.00	EA		34580.00
12	JBD203615-P84	6 RACKS - 84 X 36GB 15K RPM	2		36086.00	EA		72172.00
13	JBD201815-P14	DISK:0.3TBPDG,14X18GB,15RPM+10%SPARES	1		8772.00	EA		8772.00
14	JBD2020-SA	RCK; FC2GB STORAGE, OPTIACAL/OPTICAL I/F	13		3958.00	EA		51454.00
15	JBD2001-RMK	DISK:ENCLOSURE RACK MOUNT KIT FOR 36U RA	13		190.00	EA		2470.00
16	SFR91-PWR	ACC: 9-PLUG POWER STRIP	3		238.00	EA		714.00
17	HRT361941-OFT	ACC:36U 19X41 OPEN FRT CAB, HRT COLOR	1		1425.00	EA		1425.00
18								
19	DISCOUNT	LARGE VOLUME CASH DISCOUNT	1		-70480.00	EA		-70480.00
20								
21								
22		Disks come with a 3year						
23		return to factory warranty						
24		7 day replenishment						
25		Quote valid for 90 days						

Taxable	0.00
Tax	0.00
Exempt	471676.00
Total	471676.00
Paid	0.00
Tr Disc	0.00
Balance	471676.00

Amount shown in
Rate :0.00 of

PREFACE

Document Overview

This report documents the methodology and results of the TPC Benchmark™ H (TPC-H) test conducted on the Unisys ES7000 Aries 420 Enterprise Server using Oracle Database 10g Enterprise Edition, in conformance with the requirements of the TPC Benchmark™H Standard Specification Revision 2.0. The tests documented in this report were sponsored by Unisys Corporation. The operating system used for the benchmark was Microsoft Windows Server 2003, Datacenter Edition for 64-bit Itanium-based Systems.

The Transaction Processing Performance Council (TPC) developed the TPC-H Benchmark. The TPC Benchmark™ H Standard represents an effort by Unisys Corporation and other members of the Transaction Processing Performance Council (TPC) to create an industry-wide benchmark for evaluating the performance and price/performance of decision support systems, and to disseminate objective, verifiable performance data to the data processing industry.

A certified audit of these measurements and the reported results was performed by Lorna Livingtree of Performance Metrics Inc. (Klamath, CA). She has verified compliance with the relevant TPC Benchmark™ H specifications; audited the benchmark configuration, environment, and methodology used to produce and validate the test results; and audited the pricing model used to calculate the price/ performance. The auditor's letter of attestation is attached to the Executive Summary and precedes this section.

TPC Benchmark™H Overview

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

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

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

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

The TPC-H operations are modeled as follows:

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

The minimum database required to run the benchmark holds business data from 10,000 suppliers. It contains almost ten million rows representing a raw storage capacity of about 1 gigabyte. Compliant benchmark implementations may also use one of the larger permissible database populations (e.g., 1000 gigabytes), as defined in Clause 4.1.3.

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

The TPC-H database must be implemented using a commercially available database management system (DBMS) and the queries executed via an interface using dynamic SQL. The specification provides for variants of SQL, as implementers are not required to have implemented a specific SQL standard in full. TPC-H uses terminology and metrics that are similar to other benchmarks, originated by the TPC and others. Such similarity in terminology does not in any way imply that TPC-H results are comparable to other benchmarks. The only benchmark results comparable to TPC-H are other TPC-H results compliant with the same revision.

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

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

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

General Implementation Guidelines

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

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

A Table of Contents follows after this page.

Related Product Information

The TPC Benchmark™ H Standard requires that test sponsors provide a Full Disclosure Report in addition to published results. You can obtain copies of the test results as well as additional copies of this full disclosure report by sending a request to the following address:

Unisys Corporation
TPC Benchmark Administrator, MS 4683
Systems Analysis Modeling & Measurement
PO Box 64942
Saint Paul, MN 55164-0942

End Table of Contents

1. GENERAL ITEMS

1.1 Benchmark Sponsor

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

This TPC benchmark H was sponsored by Unisys Corporation. The benchmark test was developed by Oracle and Unisys. The benchmark was conducted at Unisys in Roseville, Minnesota.

1.2 Parameter Settings

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

- *Data Base tuning options;*
- *Optimizer/Query execution options;*
- *Query Processing tool/language configuration parameters;*
- *Recovery/commit options;*
- *Consistency/locking options;*
- *Operating system and configuration parameters;*
- *Configuration parameters and options for any other software component incorporated into the pricing structure;*
- *Compiler optimization options.*

Comment 1: In the event that some parameters and options are set multiple times, it must be easily discernible by an interested reader when the parameter or option was modified and what new value it received each time.

Comment 2: This requirement can be satisfied by providing a full list of all parameters and options, as long as all those that have been modified from their default values have been clearly identified and these parameters and options are only set once.

Details of system and database configurations and parameters are provided in Appendixes A and B.

1.3 Configuration Diagrams

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

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

The SUT had two internal disks; the priced system comes with both. The priced configuration is shown in the diagram that follows.

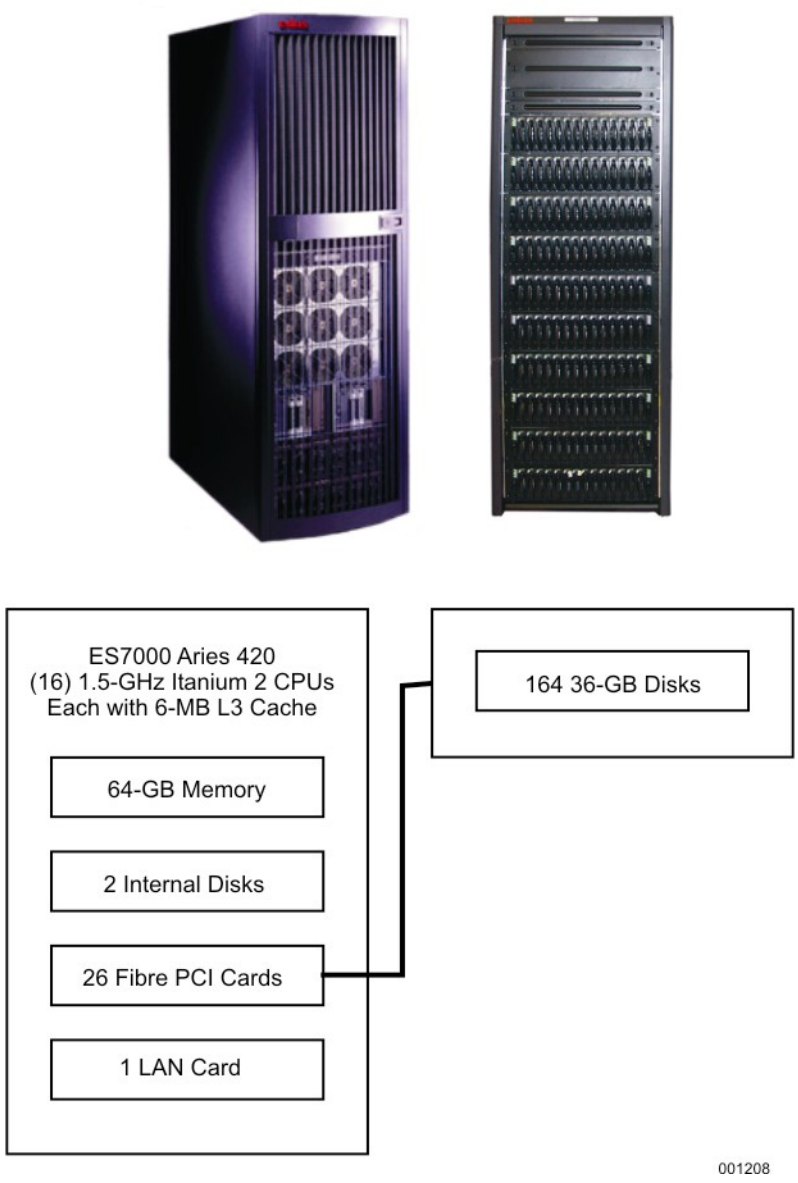


Figure 1.1 Benchmark and Priced Configuration for ES7000 Aries 420 Enterprise Server

2. CLAUSE 1: LOGICAL DATA BASE DESIGN

2.1 Table Definitions

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

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

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

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

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

No record clustering or index clustering was used. Column ordering was reordered in tables. Refer to the table create statements in Appendix B for further details.

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

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

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

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

No replication was used. See Appendix B, which contains the database and table creation statements.

2.4 Vertical Partitioning

Vertical partitioning of tables is not allowed. For example, groups of columns of one row shall not be assigned to files, disks, or areas different from those storing the other columns of that row. The row must be processed as an atomic series of contiguous columns.

Comment*: The effect of vertical partitioning is to reduce the effective row size accessed by the system. Given the synthetic nature of this benchmark, the effect of vertical partitioning is achieved by the choice of row sizes. No further vertical partitioning of the data set is allowed. Specifically, the above Clause prohibits assigning one or more of the columns not accessed by the TPC-H query set to a vertical partition.*

Vertical partitioning was not used. See Appendix B, which contains the database and table creation statements.

3. CLAUSE 2: QUERIES AND UPDATE FUNCTIONS

3.1 Query Language

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

SQL was the query language used to implement all queries.

3.2 Random Number Generation

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

DBGEN Version 1.3.0 and QGEN version 1.3.0 were used to generate random numbers for these runs.

3.3 Substitution Parameters

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

The supplied QGEN version 1.3.0 was used.

3.4 Query Text and Output Data from Qualification Database

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

Appendix C contains the query text and query output.

3.5 Query Substitution Parameters and Seeds

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

Appendix D contains the seed and query substitution parameters.

3.6 Query Isolation Level

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

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

3.7 Source Code of Refresh Functions

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

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

4. CLAUSE 3: DATABASE SYSTEM PROPERTIES

4.1 Atomicity

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

4.1.1 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 was committed.
4. The total price from the ORDER table and the extended price from the LINEITEM table were retrieved for the same order key used in Step 1. It was verified that the appropriate rows had been inserted.

4.1.2 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 Acid transaction was ROLLED BACK.
4. The total price from the ORDER table and the extended price from the LINEITEM table were retrieved for the same order key used in Step 1. It was verified that the appropriate rows had not been changed.

4.2 Consistency

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

4.2.1 Consistency Test

Verify that ORDER and LINEITEM tables are initially consistent, submit the required number of Acid transactions with randomly selected input parameters, and re-verify the consistency of the ORDER and LINEITEM tables.

The consistency of the ORDER and LINEITEM tables was verified based on randomly selected values of the column O_ORDERKEY.

1. Acid queries were executed to verify the initial consistent state of the ORDER and LINEITEM tables.
2. More than 100 Acid transactions were submitted from each of two execution streams.
3. Acid queries were re-executed to verify the consistent state of the ORDER and LINEITEM tables after the Acid transaction streams.
4. The consistency of the ORDER and LINEITEM tables was re-verified.

4.3 Isolation

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

4.3.1 Read-Write Conflict with Commit

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

1. An Acid transaction was started for a randomly selected O_KEY, L_KEY, and DELTA. The Acid transaction was suspended prior to COMMIT.

2. An ACID query was started for the same O_KEY used in Step 1. The ACID query completed and did not see the uncommitted changes made by the Acid transaction.

3. The Acid transaction was COMMITTED.

4.3.2 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 DELA. The ACID transaction was suspended prior to ROLLBACK.

2. An ACID query was started for the same O_KEY used in Step 1. The ACID query did not see the uncommitted changes made by the ACID transaction.

3. The ACID transaction was ROLLED BACK.

4. The ACID query completed.

4.3.3 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. The ACID transaction was suspended prior to COMMIT.

2. Another ACID transaction, T2, was started using the same O_KEY and L_KEY and a randomly selected DELTA.

3. T2 waited.

4. T1 was allowed to COMMIT and T2 completed.

5. It was verified that T2.L_EXTENDEDPRICE was calculated correctly.
   T2.L_EXTENDEDPRICE = T1.L_EXTENDEDPRICE +(DELTA1*(T1.L_EXTENDEDPRICE/T1.L_QUANTITY))

4.3.4 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. The Acid transaction was suspended prior to ROLLBACK.

2. Another Acid transaction, T2, was started using the same O_KEY and L_KEY and a different randomly selected DELA.

3. T2 waited

4. T1 was allowed to ROLLBACK and T2 completed

5. It was verified that T2.L_EXTENDEDPRICE = T1.L_EXTENDEDPRICE.

4.3.5 Concurrent Progress of Read and Write on Different Tables

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

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

2. Another ACID transaction, T2 was started using random values for PS_PARTKEY and PS_SUPPKEY.

3. ACID Transaction T2 completed.

4. ACID transaction T1 completed and the appropriate rows in the ORDER, LINEITEM, and HISTORY tables were changed.

4.3.6 Updates not Indefinitely Delayed by Reads on Same Table

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

1. An ACID transaction, T1, was started, executing Q1 against the qualification database. The substitution parameter was chosen from the interval [0..2159] so that the query ran for a sufficient length of time.

2. Before T1 completed, an ACID transaction, T2, was started using randomly selected values of O_KEY, L_KEY and DELTA.

3. T2 completed before T1 completed. Verified that the appropriate rows in ORDER, LINEITEM and HISTORY tables have been changed.

4.4 Durability

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

4.4.1 Failure of a Durable Medium and System Crash

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

The database tables and logs were placed on Automatic Storage Management (ASM) disk groups. The partitions were created on two logical drives of the same characteristics as the drives used for the Test database, and the two drives were on two separate controllers.

1. The data files and log files were created into one disk group using ASM's normal redundancy mirroring option.

2. Eight streams of ACID transactions were started.

3. After at least 100 transactions had occurred on each stream and the streams were still running, one ASM disk drive was removed.

4. After it was determined that the test would still run with the data and log drive removed, and after running at least another 100 transactions on each stream, the oracle database was shutdown causing a failure.

5. The eight streams of ACID transactions failed and recorded their numbers of committed transactions in success files.

6. The data and log drive was inserted back in the system.

7. The data files were restored to their state prior to the ACID transaction streams.

8. The database ran through its recovery mode.

9. The counts in the success files and the HISTORY table count were compared and the counts matched.

4.4.2 System Crash

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

The system crash and memory failure tests were combined.

1. Eight streams of ACID transactions were started.

2. After at least 100 transactions had occurred on each stream and the streams of ACID transactions were still running, the system was powered off.

3. When power was restored the system rebooted and the database was restarted.

4. The database went through a recovery period.

5. The success file and the HISTORY table counts were compared, and they matched.

4.4.3 Memory Failure

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 tests were combined. See the previous section.

5. CLAUSE 4: SCALING AND DATABASE POPULATION

5.1 Cardinality of Tables

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

TABLE	# of ROWS
Orders	1,500,000,000
Lineitem	5,999,989,709
Customer	150,000,000
Parts	200,000,000
Supplier	10,000,000
Partsupp	800,000,000
Nation	25
Region	5

5.2 Distribution of Tables and Logs Across Media

The distribution of tables and logs across all media must be explicitly described using a format similar to that shown in the following example for both the tested and priced systems.

The SUT and the priced system have 164 external drives and 2 internal drives.

Utilization of the drives:

- 156 physical/logical drives for the 1000GB database. See Appendix F for exact disk configuration.
- Two Oracle ASM diskgroups namely *dg1* and *aciddg1* were created on the SUT. *Dg1* consists of 156 external drives and *aciddg1* consists of 2 external drives.
- The Test Database, its Log and Temp files were created on ASM diskgroup *dg1*. The Test Database and its Log were mirrored by Oracle ASM. Temp was not mirrored. For ASM scripts, refer to APPENDIX F.
- The ACID database was created on Oracle ASM diskgroup *aciddg1*.
- The operating system, Microsoft Windows Server 2003, Datacenter Edition for 64-bit Itanium-based systems, and Oracle Database 10g Enterprise Edition, as well as part of the operating system page file, were installed on the internal drive.
- 6 external drives were striped and were used for update sets, benchmark output, and the bulk of the system pagefile.

5.3 Partitions/Replications Mapping

The mapping of data base partitions/replications must be explicitly described.

***Comment**: The intent is to provide sufficient detail about partitioning and replication to allow independent reconstruction of the test database.*

The database was not replicated.

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

5.4 Use of RAID

Implementations may use some form of RAID . The RAID level used must be disclosed for each device.

No hardware RAID was used in the implementation. The Test Database and its log, as well as the qualification database and its log, were mirrored using Oracle ASM. Temp was not mirrored.

5.5 DBGEN Modifications

The version number, release number, modification number, and patch level of DBGEN must be disclosed. Any modifications to the DBGEN 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 1.3.0 was used for populating the database.

5.6 Database Load Time

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

The Numerical Quantities summary (pp. v) contains the database load time, which was 5:24:42.

5.7 Data Storage Ratio

The data storage ratio must be disclosed. It is computed by dividing the total data storage of the priced configuration (expressed in GB) by the size chosen for the test database. The ratio must be reported to the nearest 1/100th, rounded up. For example, a system configured with 96 disks of 2.1 GB capacity for a 100GB test database has a data storage ratio of 2.02.

***Comment**: For the reporting of configured disk capacity, gigabyte (GB) is defined to be 2^30 bytes. Since disk manufacturers typically report disk size using base ten (i.e., GB = 10^9), it may be necessary to convert the advertised size from base ten to base two.*

Disks	# of Disks	Usable Space Per Disk	Subtotal Disk Space
Internal	2	33.47 GB	66.94 GB
External	164	34.17	5603.88 GB
		Total	5670.82 GB

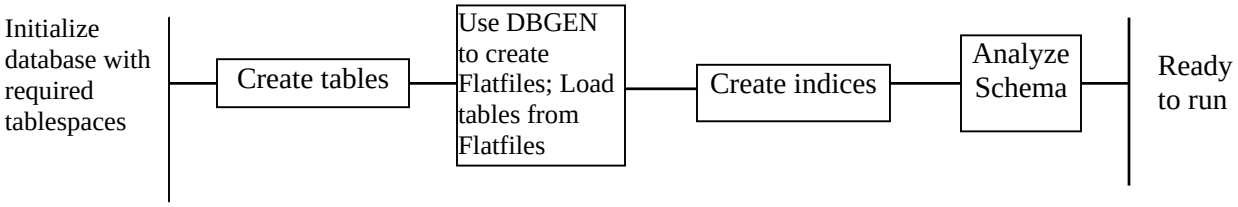
The Numerical Quantities summary (pp. v) contains the data storage ratio (5.67) for the system used.

5.8 Database Loading

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

The following steps were used to load the database:

1) DBGEN version 1.3.0 was used to create flat files.
2) The database was then loaded from those flatfiles using the scripts in Appendix B.



5.9 Qualification Database Configuration

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

The qualification database was created using scripts identical to those of the test database, except for variances due to the sizes of the two databases.

6. Clause 5: Performance Metrics and Execution Rules

6.1 System Activity Between Load and Performance Tests

Any system activity on the SUT which takes place between the conclusion of the load test and the beginning of the performance test must be fully disclosed including listings of scripts or command logs.

After the database load, the flat files were dropped and the first run begun.

6.2 Power Test Implementation

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

The following steps were followed to run the power test.

1. RF1 refresh transactions were run
2. Stream 00 execution was run
3. RF2 refresh transactions were run.

6.3 Timing Intervals and Reporting

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

This information is contained in the Numerical Quantities Summary page in the Executive Summary at the beginning of this report. For convenience, it is repeated in Section 6.10.

6.4 Number of Streams in the Throughput Test

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

Seven streams were run for the throughput test

6.5 Start and End Date/Time for Each Query Stream

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

This information is contained in the Numerical Quantities Summary page in the Executive Summary at the beginning of this report. For convenience, it is repeated in Section 6.10.

6.6 Total Elapsed Time for the Measurement Interval

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

This information is contained in the Numerical Quantities Summary page in the Executive Summary at the beginning of this report. For convenience, it is repeated in Section 6.10.

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

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

This information is contained in the Numerical Quantities Summary page in the Executive Summary at the beginning of this report. For convenience, it is repeated in Section 6.10.

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

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

This information is contained in the Numerical Quantities Summary page in the Executive Summary at the beginning of this report. For convenience, it is repeated in Section 6.10.

6.9 Performance Metrics

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

This information is contained in the Numerical Quantities Summary section of the Executive Summary (p. v in front). For convenience, it is repeated in Section 6.10.

6.10 The Performance Metric and Numerical Quantities from Both Runs

*The performance metric (QphH) and the numerical quantities (TPC-*H Power@Size and *TPC-H Throughput@Size) from both of the runs must be disclosed.*

	QppH@1000GB	**QthH@1000GB**	**QphH@1000GB**
Run 1	13629.8	7142.8	9866.9
Run 2	13614.2	7131.3	9853.3
% Difference	- 0.11%	- 0.16%	- 0.14%

(Run 2 was reported.)

Tables from Numerical Quantities pages in the front of this report, reprinted here for reader's convenience:

(see next page)

Numerical Quantities Summary

Measurement Results

Scale Factor	1000 GB
Total Data Storage / Database Size	5.67
Start of Database Load	7/17/2004 10:40:37
End of Database Load	7/17/2004 16:05:20
Database Load Time	5:24:43
Query Streams for Throughput Test	7
TPC-H Power	13614.2
TPC-H Throughput	7131.3
Composite Query per Hour Rating (QphH@1000GB)	9853.3
Total System Price Over 3 Years	$669,239
TPC-H Price Performance Metric	$67.93

Measurement Intervals

Measurement Interval in Throughput Test (Ts)	77742	seconds

Duration of Stream Execution:

		Query Start Date/Time	RF1 Start Date/Time	RF2 Start Date/Time	
	Seed	**Query End Date/Time**	**RF1 End Date/Time**	**RF2 End Date/Time**	**Duration**
Stream 0	717160520	7/18/04 20:56:33	7/18/2004 20:54:30	7/19/04 0:01:42	3:10:51
		7/19/04 0:01:42	7/18/2004 20:56:32	7/19/04 0:05:21	
Stream 1	717160521	7/19/04 0:05:40	7/19/04 20:54:58	7/19/04 20:56:58	20:54:58
		7/19/04 16:52:32	7/19/04 20:56:58	7/19/04 21:00:38	
Stream 2	717160522	7/19/04 0:05:40	7/19/04 21:00:38	7/19/04 21:02:45	21:02:41
		7/19/04 18:51:11	7/19/04 21:02:45	7/19/04 21:08:21	
Stream 3	717160523	7/19/04 0:05:41	7/19/04 21:08:21	7/19/04 21:10:32	21:08:42
		7/19/04 18:36:27	7/19/04 21:10:32	7/19/04 21:14:23	
Stream 4	717160524	7/19/04 0:05:41	7/19/04 21:14:02	7/19/04 21:16:30	21:14:20
		7/19/04 18:13:52	7/19/04 21:16:30	7/19/04 21:20:01	
Stream 5	717160525	7/19/04 0:05:41	7/19/04 21:20:14	7/19/04 21:22:24	21:22:09
		7/19/04 14:29:14	7/19/04 21:22:24	7/19/04 21:27:50	
Stream 6	717160526	7/19/04 0:05:41	7/19/04 21:27:50	7/19/04 21:30:03	21:28:05
		7/19/04 17:48:04	7/19/04 21:30:03	7/19/04 21:33:46	
Stream 7	717160527	7/19/04 0:05:41	7/19/04 21:33:46	7/19/04 21:35:58	21:35:34
		7/19/04 20:54:45	7/19/04 21:35:58	7/19/04 21:41:15	

TPC-H Timing Intervals (in seconds):

Query	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Stream 0	1067.9	117.5	79.2	84.0	334.4	67.0	420.7	242.3
Stream 1	4579.8	1595.1	591.6	502.6	2254.8	108.6	3410.0	1238.8
Stream 2	5455.5	1540.0	480.0	359.6	2155.6	1032.7	3234.4	1112.0
Stream 3	4804.2	1733.3	168.1	562.3	2392.2	166.7	3569.9	8689.4
Stream 4	5192.1	1347.1	242.7	425.1	2284.0	139.0	3149.4	863.7
Stream 5	4372.0	1484.7	327.4	165.2	2395.9	135.7	1130.1	1375.7
Stream 6	5286.4	1246.6	1609.2	467.5	1533.9	120.9	3241.5	1547.6
Stream 7	1064.2	110.3	87.8	83.0	347.3	64.1	424.3	2646.5
Min Qi	1064.2	110.3	87.8	83.0	347.3	64.1	424.3	863.7
Max Qi	5455.5	1733.3	1609.2	562.3	2395.9	1032.7	3569.9	8689.4
Avg Qi	4393.5	1293.9	501.0	366.5	1909.1	252.5	2594.2	2496.2
Query	**Q9**	**Q10**	**Q11**	**Q12**	**Q13**	**Q14**	**Q15**	**Q16**
Stream 0	1268.3	189.6	172.8	182.8	1250.9	48.5	122.9	227.9
Stream 1	6075.4	1257.8	1021.4	985.7	4319.1	362.5	622.2	1142.3
Stream 2	8217.5	1698.4	1294.6	1394.1	2780.5	203.0	530.3	1092.1
Stream 3	7158.1	804.9	841.0	310.3	3515.8	230.3	757.6	458.6
Stream 4	7167.2	1243.8	973.9	1599.8	5384.1	278.3	400.6	732.3
Stream 5	5094.1	895.4	942.2	2601.1	3983.6	432.4	183.9	538.8
Stream 6	5455.0	1883.8	816.8	2040.3	5761.3	281.7	1860.3	1199.0
Stream 7	1065.5	172.0	174.5	272.8	1251.8	47.9	121.7	231.3
Min Qi	1065.5	172.0	174.5	272.8	1251.8	47.9	121.7	231.3
Max Qi	8217.5	1883.8	1294.6	2601.1	5761.3	432.4	1860.3	1199.0
Avg Qi	5747.5	1136.6	866.3	1314.9	3856.6	262.3	639.5	770.6
Query	**Q17**	**Q18**	**Q19**	**Q20**	**Q21**	**Q22**	**RF1**	**RF2**
Stream 0	372.7	2146.1	460.1	156.3	1908.1	189.2	122.5	218.5
Stream 1	929.1	19441.8	1027.6	2927.9	5193.7	824.0	120.0	220.1
Stream 2	7314.4	14830.0	1342.9	881.7	9368.2	1212.6	126.7	335.8
Stream 3	952.7	15024.5	1461.4	921.1	11375.4	748.7	131.3	230.7
Stream 4	733.8	14056.8	985.9	429.8	17198.2	463.6	127.1	224.1
Stream 5	740.4	16434.2	1045.5	1155.5	5235.5	1144.4	129.5	326.4
Stream 6	1077.8	14981.6	1650.6	917.6	9805.6	958.0	132.3	223.0
Stream 7	384.9	56289.5	464.7	1415.3	8035.8	189.4	132.1	316.6
Min Qi	384.9	14056.8	464.7	429.8	5193.7	189.4	120.0	220.1
Max Qi	7314.4	56289.5	1650.6	2927.9	17198.2	1212.6	132.3	335.8
Avg Qi	1733.3	21579.8	1139.8	1235.6	9458.9	791.5	128.4	268.1

6.11 System Activity Between Tests

Any activity on the SUT that takes place between the conclusion of Run1 and the beginning of Run2 must be fully disclosed including listings of scripts or command logs along with any system reboots or database restarts.

The only activity which took place between the conclusion of Run 1 and the beginning of Run 2 was the capturing of the throughput test timing from the console.

7 Clause 6 SUT and Driver Implementation

7.1 Driver

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

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

For each power-test run:

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

7.2 Implementation-Specific Layer

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

The source code for the “qexec” Utility can be found in Appendix E.

8. Clause 7: Pricing Related Items

8.1 Hardware and Software Used

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

A detail list of hardware and software components, along with their style numbers and prices, is given in the Executive Summary, page iv, and in the pricing quotes on pages ix and x.

8.2 Three-Year Cost of System Configuration

The total 3-year price of the entire configuration must be reported, including: hardware, software, and maintenance charges. Separate component pricing is required.

The pricing summary sheet on page iv in the front of this report contains all details.

8.3 Availability Dates

The committed delivery date for general availability (availability date) of products used in the priced calculations must be reported. When the priced system includes products with different availability dates, the single availability date reported on the first page of the executive summary must be the date by which all components are committed to being available. The full disclosure report must report availability dates individually for at least each of the categories for which a pricing subtotal must be provided (see Clause 7.3.1.4). All availability dates, whether for individual components or for the SUT as a whole, must be disclosed to a precision of 1 day, but the precise format is left to the test sponsor.

Summary by category from the measured and priced configuration:

Category	**Available**
Server Hardware	Now (date of publication)
Storage	Now (date of publication)
Server Software except for Oracle DB 10g on Win-64	Now (date of publication)
Oracle DB 10g on Win-64	November 30, 2004

9. Clause 8: Audit Related Items

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

Lorna Livingtree of Performance, a certified TPC-H auditor, audited this benchmark

Lorna Livingtree

Performance Metrics Inc.

PO Box 984

Klamath, CA 95548

707-482-0523 Fax: 707-482-0575

See pages vii-viii in the front of this paper for a copy of the auditor's attestation letter.

Further information regarding the audit process may be obtained from Ms. Livingtree.

APPENDIX A: System and Database Tunable Parameters

Software levels:

Microsoft Windows Server 2003, Datacenter Edition
Oracle Database 10g Enterprise Edition 10.1.0.2.0

```
C:\Documents and Settings\Administrator>systeminfo

Host Name:                 LE334P1
OS Name:                   Microsoft(R) Windows(R) Server 2003, Datacenter Edition
OS Version:                5.2.3790 Build 3790
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Server
OS Build Type:             Multiprocessor Free
Registered Owner:          SAMM
Registered Organization:   UNISYS
Product ID:                69769-OEM-4216807-00054
Original Install Date:     9/24/2003, 4:22:27 PM
System Up Time:            0 Days, 16 Hours, 21 Minutes, 29 Seconds
System Manufacturer:       UNISYS
System Model:              ES7000
System Type:               Itanium (TM) -based System
Processor(s):              16 Processor(s) Installed.
Processor                  [01]: ia64 Family 31 Model 1 Stepping 5 GenuineIntel
~1500 Mhz
Processor                  [02]: ia64 Family 31 Model 1 Stepping 5 GenuineIntel
~1500 Mhz
Processor                  [03]: ia64 Family 31 Model 1 Stepping 5 GenuineIntel
~1500 Mhz
Processor                  [04]: ia64 Family 31 Model 1 Stepping 5 GenuineIntel
~1500 Mhz
Processor                  [05]: ia64 Family 31 Model 1 Stepping 5 GenuineIntel
~1500 Mhz
Processor                  [06]: ia64 Family 31 Model 1 Stepping 5 GenuineIntel
~1500 Mhz
Processor                  [07]: ia64 Family 31 Model 1 Stepping 5 GenuineIntel
~1500 Mhz
Processor                  [08]: ia64 Family 31 Model 1 Stepping 5 GenuineIntel
~1500 Mhz
Processor                  [09]: ia64 Family 31 Model 1 Stepping 5 GenuineIntel
~1500 Mhz
Processor                  [10]: ia64 Family 31 Model 1 Stepping 5 GenuineIntel
~1500 Mhz
Processor                  [11]: ia64 Family 31 Model 1 Stepping 5 GenuineIntel
~1500 Mhz
Processor                  [12]: ia64 Family 31 Model 1 Stepping 5 GenuineIntel
~1500 Mhz
Processor                  [13]: ia64 Family 31 Model 1 Stepping 5 GenuineIntel
~1500 Mhz
Processor                  [14]: ia64 Family 31 Model 1 Stepping 5 GenuineIntel
~1500 Mhz
Processor                  [15]: ia64 Family 31 Model 1 Stepping 5 GenuineIntel
~1500 Mhz
Processor                  [16]: ia64 Family 31 Model 1 Stepping 5 GenuineIntel
~1500 Mhz
Copyright 1n:              BIOS Release 2.1.001
Windows Directory:         C:\WINDOWS
System Directory:          C:\WINDOWS\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-us;English (United States)
Input Locale:              en-us;English (United States)
```

```
Time Zone:                 (GMT-06:00) Central Time (US & Canada)
Total Physical Memory:     65,526 MB
Available Physical Memory: 61,271 MB
Page File: Max Size:       194,642 MB
Page File: Available:      182,280 MB
Page File: In Use:         12,362 MB
Page File Location(s):     c:\pagefile.sys
                           u:\pagefile.sys
Domain:                    WORKGROUP
Hotfix(s):                 29 Hotfix(s) Installed.
                           [01]: File 1
                           [02]: File 1
                           [03]: File 1
                           [04]: File 1
                           [05]: File 1
                           [06]: File 1
                           [07]: File 1
                           [08]: File 1
                           [09]: File 1
                           [10]: File 1
                           [11]: File 1
                           [12]: File 1
                           [13]: File 1
                           [14]: File 1
                           [15]: Q147222
                           [16]: KB819696 - Update
                           [17]: KB822925 - Update
                           [18]: KB823182 - Update
                           [19]: KB823559 - Update
                           [20]: KB823980 - Update
                           [21]: KB824105 - Update
                           [22]: KB824141 - Update
                           [23]: KB824145 - Update
                           [24]: KB824146 - Update
                           [25]: KB825119 - Update
                           [26]: KB828028 - Update
                           [27]: KB828035 - Update
                           [28]: KB832483 - Update
                           [29]: KB832894 - Update
Network Card(s):           1 NIC(s) Installed.
                           [01]: Intel(R) PRO/1000 XT Server Adapter
                                 Connection Name: Local Area Connection 6
                                 DHCP Enabled:    No
```

Oracle Database 10g Enterprise Edition Installation

Oracle Database 10g Enterprise Edition was installed on the SUT. All default options were selected during the install except:

- Services > OracleServiceTPCH > Use Administrator Account Authentication Mode

Oracle Database 10g Enterprise Edition Parameters

Inittpch.ora

```
INSTANCE_TYPE= RDBMS
DB_CREATE_FILE_DEST= '+dg1'
db_name= tpch
undo_management= auto
control_files= '+dg1/tpch/CONTROLFILE/Current.256.1'

#control_files= '+dg1'
statistics_level                = BASIC
#statistics_level                = typical
audit_trail                     = FALSE
compatible                      = 10.1.0.2
db_block_checksum               = false
db_block_size                   = 8192
db_cache_size                   = 4g
aq_tm_processes           = 0
db_file_multiblock_read_count   = 128
db_files                        = 512
db_name                         = tpch
db_writer_processes             = 2
dml_locks                       = 40000
enqueue_resources               = 40000
global_names                    = FALSE
shared_pool_size                = 1G
large_pool_size                 = 1G
#log_buffer                      = 33554432
log_buffer                      = 4194304
log_checkpoints_to_alert        = true
max_dump_file_size              = unlimited
timed_statistics                = false
nls_date_format                 = YYYY-MM-DD
open_cursors                    = 600
optimizer_index_cost_adj        = 25
optimizer_mode                  = CHOOSE
optimizer_features_enable       = 10.1.0.1
parallel_adaptive_multi_user    = TRUE
parallel_execution_message_size = 16384
parallel_max_servers            = 256
parallel_min_servers            = 256
pga_aggregate_target            = 24G
processes                       = 500
query_rewrite_enabled           = true
recovery_parallelism            = 8
```

```
sessions                          = 500
transactions                      = 10
undo_management                   = auto
undo_retention                    = 100000
remote_login_passwordfile       = none
BACKGROUND_DUMP_DEST= C:\ORACLE\ADMIN\TPCH\BDUMP
CORE_DUMP_DEST= C:\ORACLE\ADMIN\TPCH\CDUMP
USER_DUMP_DEST= C:\ORACLE\ADMIN\TPCH\UDUMP
```

Init+data.ora

```
INSTANCE_TYPE = ASM
db_unique_name = +data
ASM_DISKSTRING='c:\asm\disk*'
processes = 200

remote_login_passwordfile=SHARED
background_dump_dest=C:\ORACLE\ADMIN\+data\bdump
core_dump_dest=C:\ORACLE\ADMIN\+data\cdump
user_dump_dest=C:\ORACLE\ADMIN\+data\udump
```

APPENDIX B: Database, Tables, and Indexes Creation

Create database

DBCRE.sh

```
#!/bin/ksh
sqlplus -s /NOLOG << !
connect / as sysdba

host date
shutdown abort
startup pfile= c:\oracle\ora10\database\inittpch.ora nomount;
create database
  controlfile reuse
  logfile '+dg1' size 10G reuse,
          '+dg1' size 10G reuse
  datafile '+dg1' size 2G EXTENT MANAGEMENT LOCAL
  sysaux datafile '+dg1' size 1G reuse
  undo tablespace ts_undo1
      datafile  '+dg1' size 20G reuse
  default temporary tablespace ts_temp
      tempfile '+dg1' size 24G reuse
      extent management local uniform size 30m
  maxdatafiles 1000
;

spool c:\tpch\dbcre\catlog100g.txt
@c:\oracle\ora10\rdbms\admin\catalog.sql;
@c:\oracle\ora10\rdbms\admin\catparr.sql;
@c:\oracle\ora10\rdbms\admin\catproc.sql;
connect system/manager
@c:\oracle\ora10\rdbms\admin\utlxplan.sql;
@c:\oracle\ora10\sqlplus\admin\pupbld.sql;
spool off
exit
!
```

Sctso.sh

```
#!/bin/ksh
echo starting sctso at `date`
sh ts_default.sh
wait
echo starting lineitem at `date`
sh ts_lineitemR1.sh
sh ts_LineitemR2.sh &
sh ts_LineitemR3.sh &
sh ts_LineitemR4.sh &
sh ts_LineitemR5.sh &
wait
echo finished lineitem at `date`
echo
echo
echo starting order at `date`
sh ts_ordersR1.sh
sh ts_ordersR2.sh &
```

```
sh ts_ordersR3.sh &
wait
echo finished order at `date`echo
echo
echo starting psupp at `date`
sh ts_psuppR1.sh
sh ts_psuppR2.sh &
sh ts_psuppR3.sh &
wait
echo finished psupp at `date`echo
echo
echo starting rest at `date`
sh ts_restR1.sh
sh ts_restR2.sh &
sh ts_restR3.sh &
wait
echo finished rest at `date`echo
echo
echo starting index at `date`
sh ts_indexR1.sh
sh ts_indexR2.sh &
sh ts_indexR3.sh &
wait
echo finished index at `date`echo
echo
echo starting temp at `date`
sh ts_tempR1.sh &
sh ts_tempR2.sh &
wait
echo ending sctsco at `date`
```

ts_default.sh

```
#!/bin/ksh
sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
drop tablespace ts_default including contents;
create tablespace ts_default
datafile '+dg1' size 1G reuse
extent management local
autoallocate
;
exit
!
```

ts_indexR1.sh

```
#!/bin/ksh
sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
drop tablespace ts_index including contents;
create tablespace ts_index datafile
size 21G
;
exit
!
```

ts_indexR2.sh

```
#!/bin/ksh
sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
alter tablespace ts_index add datafile
```

```
size 21G, size 21G, size 21G, size 21G,
size 21G, size 21G, size 21G, size 21G
;
exit
!
```

ts_indexR3.sh

```
#!/bin/ksh
sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
alter tablespace ts_index add datafile
size 21G, size 21G, size 21G, size 21G,
size 21G, size 21G, size 21G
;
exit
!
```

ts_lineitemR1.sh

```
#!/bin/ksh
sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
drop tablespace ts_l1 including contents;
create tablespace ts_l1 datafile
size 32000M
;
exit
!
```

ts_lineitemR2.sh

```
#!/bin/ksh
sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
alter tablespace ts_l1 add datafile
size 32000M, size 32000M, size 32000M, size 32000M,
size 32000M, size 32000M, size 32000M, size 32000M
;
exit
!
```

ts_lineitemR3.sh

```
#!/bin/ksh
sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
alter tablespace ts_l1 add datafile
size 32000M, size 32000M, size 32000M, size 32000M,
size 32000M, size 32000M, size 32000M, size 32000M
;
exit
!
```

ts_lineitemR4.sh

```
#!/bin/ksh
sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
alter tablespace ts_l1 add datafile
```

```
size 32000M, size 32000M, size 32000M, size 32000M,
size 32000M, size 32000M, size 32000M, size 32000M
;
exit
!
```

ts_lineitemR5.sh

```
#!/bin/ksh
sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
alter tablespace ts_l1 add datafile
size 32000M, size 32000M, size 32000M, size 32000M,
size 32000M, size 32000M, size 32000M
;
exit
!
```

ts_ordersR1.sh

```
#!/bin/ksh
sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
drop tablespace ts_o1 including contents;
create tablespace ts_o1 datafile
size 16G
;
exit
!
```

ts_ordersR2.sh

```
#!/bin/ksh

sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
alter tablespace ts_o1 add datafile
size 16G, size 16G, size 16G, size 16G,
size 16G, size 16G, size 16G, size 16G
;
exit
!
```

ts_ordersR3.sh

```
#!/bin/ksh

sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
alter tablespace ts_o1 add datafile
size 16G, size 16G, size 16G, size 16G, size 16G, size 16G, size 16G
;

exit
!
```

ts_psuppR1.sh

```
#!/bin/ksh
sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
```

```
drop tablespace ts_psupp including contents;
create tablespace ts_psupp datafile
size 14G
;
exit
!
```

ts_psuppR2.sh

```
#!/bin/ksh

sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
alter tablespace ts_psupp add datafile
size 14G, size 14G, size 14G, size 14G,
size 14G, size 14G, size 14G, size 14G
;

exit
!
```

ts_psuppR3.sh

```
#!/bin/ksh

sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
alter tablespace ts_psupp add datafile
size 14G, size 14G, size 14G, size 14G,
size 14G, size 14G, size 14G
;

exit
!
```

ts_restR1.sh

```
#!/bin/ksh
sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
drop tablespace ts_rest including contents;
create tablespace ts_rest datafile
size 7G
;
exit
!
```

ts_restR2.sh

```
#!/bin/ksh
sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
alter tablespace ts_rest add datafile
size 7G, size 7G, size 7G, size 7G,
size 7G, size 7G, size 7G, size 7G
;
exit
!
```

ts_restR3.sh

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

```
sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
alter tablespace ts_rest add datafile
size 7G, size 7G, size 7G, size 7G,
size 7G, size 7G, size 7G
;
exit
!
```

ts_tempR1.sh

```
#!/bin/ksh
sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
alter tablespace ts_temp add tempfile
size 24G, size 24G, size 24G, size 24G,
size 24G, size 24G, size 24G, size 24G,
size 24G, size 24G, size 24G, size 24G,
size 24G, size 24G, size 24G, size 24G
;
exit
!
```

ts_tempR2.sh

```
#!/bin/ksh
sqlplus -s /NOLOG << !
connect / as sysdba
set timing on
alter tablespace ts_temp add tempfile
size 24G, size 24G, size 24G, size 24G,
size 24G, size 24G, size 24G, size 24G,
size 24G, size 24G, size 24G, size 24G,
size 24G, size 24G, size 24G
;
exit
!
```

DBLOAD.sh

```
#!/bin/ksh
sqlplus -s /NOLOG << !
connect / as sysdba
shutdown abort;
startup;

host date

spool dapop.txt

set timing on

grant DBA
to tpch identified by tpch;

connect tpch/tpch;
drop directory data_dir;

create directory data_dir as 'F:/';

drop table l_et;
```

```
create table l_et(
   l_shipdate          date ,
   l_orderkey          number ,
   l_discount          number ,
   l_extendedprice     number ,
   l_suppkey           number ,
   l_quantity          number ,
   l_returnflag        char(1) ,
   l_partkey           number ,
   l_linestatus        char(1) ,
   l_tax               number ,
   l_commitdate        date ,
   l_receiptdate       date ,
   l_shipmode          char(10) ,
   l_linenumber        number ,
   l_shipinstruct      char(25) ,
   l_comment           varchar(44)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                    records delimited by newline
                    nobadfile
                    nologfile
                    fields terminated by '|'
                    missing field values are null
          )
          location (
'lineitem.tbl.1','lineitem.tbl.2','lineitem.tbl.3','lineitem.tbl.4',
'lineitem.tbl.5','lineitem.tbl.6','lineitem.tbl.7','lineitem.tbl.8',
'lineitem.tbl.9','lineitem.tbl.10','lineitem.tbl.11','lineitem.tbl.12',
'lineitem.tbl.13','lineitem.tbl.14','lineitem.tbl.15', 'lineitem.tbl.16'
))
reject limit unlimited;

drop table o_et;
create table o_et(
   o_orderdate         date ,
   o_orderkey          number ,
   o_custkey           number ,
   o_orderpriority     char(15) ,
   o_shippriority      number ,
   o_clerk             char(15) ,
   o_orderstatus       char(1) ,
   o_totalprice        number ,
   o_comment           varchar(79)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                    records delimited by newline
                    nobadfile
                    nologfile
                    fields terminated by '|'
                    missing field values are null
          )
          location (
'orders.tbl.1','orders.tbl.2','orders.tbl.3','orders.tbl.4',
'orders.tbl.5','orders.tbl.6','orders.tbl.7','orders.tbl.8',
'orders.tbl.9','orders.tbl.10','orders.tbl.11','orders.tbl.12',
```

```
'orders.tbl.13','orders.tbl.14','orders.tbl.15','orders.tbl.16'
))
reject limit unlimited;


drop table ps_et;
create table ps_et(
    ps_partkey          number ,
    ps_suppkey          number ,
    ps_supplycost       number ,
    ps_availqty         number ,
    ps_comment          varchar(199)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
            )
            location (
'partsupp.tbl.1','partsupp.tbl.2','partsupp.tbl.3','partsupp.tbl.4',
'partsupp.tbl.5','partsupp.tbl.6','partsupp.tbl.7','partsupp.tbl.8',
'partsupp.tbl.9','partsupp.tbl.10','partsupp.tbl.11','partsupp.tbl.12',
'partsupp.tbl.13','partsupp.tbl.14','partsupp.tbl.15','partsupp.tbl.16'
))
reject limit unlimited;


drop table p_et;
create table p_et(
    p_partkey           number ,
    p_type              varchar(25) ,
    p_size              number ,
    p_brand             char(10) ,
    p_name              varchar(55) ,
    p_container         char(10) ,
    p_mfgr              char(25) ,
    p_retailprice       number ,
    p_comment           varchar(23)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
            )
            location (
'part.tbl.1', 'part.tbl.2', 'part.tbl.3', 'part.tbl.4',
'part.tbl.5', 'part.tbl.6', 'part.tbl.7', 'part.tbl.8',
'part.tbl.9', 'part.tbl.10', 'part.tbl.11', 'part.tbl.12',
'part.tbl.13', 'part.tbl.14', 'part.tbl.15', 'part.tbl.16'
))
reject limit unlimited;
```

```
drop table c_et;
create table c_et(
    c_custkey           number ,
    c_mktsegment          char(10) ,
    c_nationkey         number ,
    c_name              varchar(25) ,
    c_address           varchar(40) ,
    c_phone             char(15) ,
    c_acctbal           number ,
    c_comment             varchar(117)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
            )
            location (
'customer.tbl.1', 'customer.tbl.2', 'customer.tbl.3', 'customer.tbl.4',
'customer.tbl.5', 'customer.tbl.6', 'customer.tbl.7', 'customer.tbl.8',
'customer.tbl.9', 'customer.tbl.10', 'customer.tbl.11', 'customer.tbl.12',
'customer.tbl.13', 'customer.tbl.14', 'customer.tbl.15', 'customer.tbl.16'
))
reject limit unlimited;


drop table s_et;
create table s_et(
    s_suppkey           number ,
    s_nationkey         number ,
    s_comment             varchar(101) ,
    s_name              char(25) ,
    s_address           varchar(40) ,
    s_phone             char(15) ,
    s_acctbal           number
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
            )
location (
'supplier.tbl.1', 'supplier.tbl.2', 'supplier.tbl.3', 'supplier.tbl.4',
'supplier.tbl.5', 'supplier.tbl.6', 'supplier.tbl.7', 'supplier.tbl.8',
'supplier.tbl.9', 'supplier.tbl.10', 'supplier.tbl.11', 'supplier.tbl.12',
'supplier.tbl.13', 'supplier.tbl.14', 'supplier.tbl.15', 'supplier.tbl.16'
))
reject limit unlimited;


drop table n_et;
```

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



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



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


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

@c:/ORACLE/ORA10/rdbms/admin/utlxplan.sql;


set echo on
set timing on
```

```
drop table lineitem;
create table lineitem(
   l_shipdate          ,
   l_orderkey          NOT NULL,
   l_discount          NOT NULL,
   l_extendedprice     NOT NULL,
   l_suppkey           NOT NULL,
   l_quantity          NOT NULL,
   l_returnflag        ,
   l_partkey           NOT NULL,
   l_linestatus        ,
   l_tax               NOT NULL,
   l_commitdate        ,
   l_receiptdate       ,
   l_shipmode          ,
   l_linenumber        NOT NULL,
   l_shipinstruct      ,
   l_comment
)
pctfree 1
pctused 99
initrans 10
storage (freelists 99)
parallel
nologging
partition by range (l_shipdate)
subpartition by hash(l_partkey)
subpartitions 32
(
partition item1 values less than (to_date('1992-01-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item2 values less than (to_date('1992-02-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item3 values less than (to_date('1992-03-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item4 values less than (to_date('1992-04-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item5 values less than (to_date('1992-05-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item6 values less than (to_date('1992-06-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item7 values less than (to_date('1992-07-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item8 values less than (to_date('1992-08-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item9 values less than (to_date('1992-09-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item10 values less than (to_date('1992-10-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item11 values less than (to_date('1992-11-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item12 values less than (to_date('1992-12-01','YYYY-MM-DD'))
store in (ts_l1)
```

```
,
partition item13 values less than (to_date('1993-01-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item14 values less than (to_date('1993-02-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item15 values less than (to_date('1993-03-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item16 values less than (to_date('1993-04-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item17 values less than (to_date('1993-05-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item18 values less than (to_date('1993-06-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item19 values less than (to_date('1993-07-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item20 values less than (to_date('1993-08-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item21 values less than (to_date('1993-09-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item22 values less than (to_date('1993-10-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item23 values less than (to_date('1993-11-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item24 values less than (to_date('1993-12-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item25 values less than (to_date('1994-01-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item26 values less than (to_date('1994-02-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item27 values less than (to_date('1994-03-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item28 values less than (to_date('1994-04-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item29 values less than (to_date('1994-05-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item30 values less than (to_date('1994-06-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item31 values less than (to_date('1994-07-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item32 values less than (to_date('1994-08-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item33 values less than (to_date('1994-09-01','YYYY-MM-DD'))
store in (ts_l1)
,
```

```
partition item34 values less than (to_date('1994-10-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item35 values less than (to_date('1994-11-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item36 values less than (to_date('1994-12-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item37 values less than (to_date('1995-01-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item38 values less than (to_date('1995-02-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item39 values less than (to_date('1995-03-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item40 values less than (to_date('1995-04-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item41 values less than (to_date('1995-05-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item42 values less than (to_date('1995-06-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item43 values less than (to_date('1995-07-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item44 values less than (to_date('1995-08-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item45 values less than (to_date('1995-09-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item46 values less than (to_date('1995-10-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item47 values less than (to_date('1995-11-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item48 values less than (to_date('1995-12-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item49 values less than (to_date('1996-01-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item50 values less than (to_date('1996-02-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item51 values less than (to_date('1996-03-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item52 values less than (to_date('1996-04-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item53 values less than (to_date('1996-05-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item54 values less than (to_date('1996-06-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item55 values less than (to_date('1996-07-01','YYYY-MM-DD'))
```

```
store in (ts_l1)
,
partition item56 values less than (to_date('1996-08-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item57 values less than (to_date('1996-09-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item58 values less than (to_date('1996-10-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item59 values less than (to_date('1996-11-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item60 values less than (to_date('1996-12-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item61 values less than (to_date('1997-01-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item62 values less than (to_date('1997-02-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item63 values less than (to_date('1997-03-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item64 values less than (to_date('1997-04-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item65 values less than (to_date('1997-05-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item66 values less than (to_date('1997-06-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item67 values less than (to_date('1997-07-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item68 values less than (to_date('1997-08-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item69 values less than (to_date('1997-09-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item70 values less than (to_date('1997-10-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item71 values less than (to_date('1997-11-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item72 values less than (to_date('1997-12-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item73 values less than (to_date('1998-01-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item74 values less than (to_date('1998-02-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item75 values less than (to_date('1998-03-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item76 values less than (to_date('1998-04-01','YYYY-MM-DD'))
store in (ts_l1)
```

```
,
partition item77 values less than (to_date('1998-05-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item78 values less than (to_date('1998-06-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item79 values less than (to_date('1998-07-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item80 values less than (to_date('1998-08-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item81 values less than (to_date('1998-09-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item82 values less than (to_date('1998-10-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item83 values less than (to_date('1998-11-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item84 values less than (MAXVALUE)
store in (ts_l1))
as select
   l_shipdate          ,
   l_orderkey          ,
   l_discount          ,
   l_extendedprice     ,
   l_suppkey           ,
   l_quantity          ,
   l_returnflag        ,
   l_partkey           ,
   l_linestatus        ,
   l_tax               ,
   l_commitdate        ,
   l_receiptdate       ,
   l_shipmode          ,
   l_linenumber        ,
   l_shipinstruct      ,
   l_comment
from l_et;



drop table orders;
create table orders(
   o_orderdate         ,
   o_orderkey          NOT NULL,
   o_custkey           NOT NULL,
   o_orderpriority     ,
   o_shippriority      ,
   o_clerk             ,
   o_orderstatus       ,
   o_totalprice        ,
   o_comment
)
pctfree 1
pctused 99
initrans 10
storage (freelists 99)
parallel
nologging
partition by range (o_orderdate)
```

```
subpartition by hash(o_custkey)
subpartitions 32
(
partition ord1 values less than (to_date('1992-01-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord2 values less than (to_date('1992-02-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord3 values less than (to_date('1992-03-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord4 values less than (to_date('1992-04-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord5 values less than (to_date('1992-05-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord6 values less than (to_date('1992-06-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord7 values less than (to_date('1992-07-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord8 values less than (to_date('1992-08-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord9 values less than (to_date('1992-09-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord10 values less than (to_date('1992-10-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord11 values less than (to_date('1992-11-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord12 values less than (to_date('1992-12-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord13 values less than (to_date('1993-01-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord14 values less than (to_date('1993-02-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord15 values less than (to_date('1993-03-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord16 values less than (to_date('1993-04-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord17 values less than (to_date('1993-05-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord18 values less than (to_date('1993-06-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord19 values less than (to_date('1993-07-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord20 values less than (to_date('1993-08-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord21 values less than (to_date('1993-09-01','YYYY-MM-DD'))
```

```
store in (ts_o1)
,
partition ord22 values less than (to_date('1993-10-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord23 values less than (to_date('1993-11-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord24 values less than (to_date('1993-12-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord25 values less than (to_date('1994-01-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord26 values less than (to_date('1994-02-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord27 values less than (to_date('1994-03-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord28 values less than (to_date('1994-04-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord29 values less than (to_date('1994-05-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord30 values less than (to_date('1994-06-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord31 values less than (to_date('1994-07-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord32 values less than (to_date('1994-08-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord33 values less than (to_date('1994-09-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord34 values less than (to_date('1994-10-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord35 values less than (to_date('1994-11-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord36 values less than (to_date('1994-12-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord37 values less than (to_date('1995-01-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord38 values less than (to_date('1995-02-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord39 values less than (to_date('1995-03-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord40 values less than (to_date('1995-04-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord41 values less than (to_date('1995-05-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord42 values less than (to_date('1995-06-01','YYYY-MM-DD'))
store in (ts_o1)
```

```
,
partition ord43 values less than (to_date('1995-07-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord44 values less than (to_date('1995-08-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord45 values less than (to_date('1995-09-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord46 values less than (to_date('1995-10-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord47 values less than (to_date('1995-11-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord48 values less than (to_date('1995-12-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord49 values less than (to_date('1996-01-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord50 values less than (to_date('1996-02-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord51 values less than (to_date('1996-03-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord52 values less than (to_date('1996-04-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord53 values less than (to_date('1996-05-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord54 values less than (to_date('1996-06-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord55 values less than (to_date('1996-07-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord56 values less than (to_date('1996-08-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord57 values less than (to_date('1996-09-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord58 values less than (to_date('1996-10-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord59 values less than (to_date('1996-11-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord60 values less than (to_date('1996-12-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord61 values less than (to_date('1997-01-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord62 values less than (to_date('1997-02-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord63 values less than (to_date('1997-03-01','YYYY-MM-DD'))
store in (ts_o1)
,
```

```
partition ord64 values less than (to_date('1997-04-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord65 values less than (to_date('1997-05-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord66 values less than (to_date('1997-06-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord67 values less than (to_date('1997-07-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord68 values less than (to_date('1997-08-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord69 values less than (to_date('1997-09-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord70 values less than (to_date('1997-10-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord71 values less than (to_date('1997-11-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord72 values less than (to_date('1997-12-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord73 values less than (to_date('1998-01-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord74 values less than (to_date('1998-02-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord75 values less than (to_date('1998-03-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord76 values less than (to_date('1998-04-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord77 values less than (to_date('1998-05-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord78 values less than (to_date('1998-06-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord79 values less than (to_date('1998-07-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord80 values less than (to_date('1998-08-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord81 values less than (to_date('1998-09-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord82 values less than (to_date('1998-10-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord83 values less than (to_date('1998-11-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord84 values less than (MAXVALUE)
store in (ts_o1)
)
as select
```

```
    o_orderdate         ,
    o_orderkey          ,
    o_custkey           ,
    o_orderpriority     ,
    o_shippriority      ,
    o_clerk             ,
    o_orderstatus       ,
    o_totalprice        ,
    o_comment
from o_et;


connect tpch/tpch
set timing on
set echo on


drop table partsupp;
create table partsupp(
    ps_partkey          NOT NULL,
    ps_suppkey          NOT NULL,
    ps_supplycost       NOT NULL,
    ps_availqty         ,
    ps_comment          ,
constraint pk_partkey_suppkey_1 primary key(ps_partkey, ps_suppkey)
)
organization index
partition by hash(ps_partkey)
partitions 32
parallel
nologging
pctthreshold 50
tablespace ts_psupp
as select
    ps_partkey          ,
    ps_suppkey          ,
    ps_supplycost       ,
    ps_availqty         ,
    ps_comment
from ps_et;

drop table customer;
create table customer(
    c_custkey           NOT NULL,
    c_mktsegment        ,
    c_nationkey         ,
    c_name              ,
    c_address           ,
    c_phone             ,
    c_acctbal           ,
    c_comment
)
pctfree 0
pctused 99
parallel
nologging
partition by hash (c_custkey)
partitions 32
tablespace ts_rest
as select
    c_custkey           ,
    c_mktsegment        ,
    c_nationkey         ,
    c_name              ,
```

```
   c_address      ,
   c_phone        ,
   c_acctbal      ,
   c_comment
from c_et;
!date

drop table part;

create table part(
   p_partkey      NOT NULL,
   p_type         ,
   p_size         ,
   p_brand        ,
   p_name         ,
   p_container    ,
   p_mfgr         ,
   p_retailprice  ,
   p_comment
)
pctfree 0
pctused 99
parallel
nologging
partition by hash (p_partkey)
partitions 32
tablespace ts_rest
as select
   p_partkey      ,
   p_type         ,
   p_size         ,
   p_brand        ,
   p_name         ,
   p_container    ,
   p_mfgr         ,
   p_retailprice  ,
   p_comment
from p_et;

drop table supplier;
create table supplier(
   s_suppkey      NOT NULL,
   s_nationkey    ,
   s_comment      ,
   s_name         ,
   s_address      ,
   s_phone        ,
   s_acctbal
)
pctfree 0
pctused 99
parallel
nologging
partition by hash (s_suppkey)
partitions 32
tablespace ts_rest
as select
   s_suppkey      ,
   s_nationkey    ,
   s_comment      ,
   s_name         ,
   s_address      ,
   s_phone        ,
   s_acctbal
```

```
from s_et;

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

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

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

host date
spool off;

connect tpch/tpch;

spool ixcre.out
host date
set echo on
set timing on

drop index i_l_orderkey;
create index i_l_orderkey
on lineitem (l_orderkey)
pctfree 5
initrans 10
tablespace ts_index
storage (freelist groups 4 freelists 84)
parallel
compute statistics
nologging;


drop index i_o_orderkey;
create unique index i_o_orderkey
on orders (o_orderkey)
pctfree 5
initrans 10
tablespace ts_index
storage (freelist groups 4 freelists 84 )
parallel
compute statistics
nologging;

drop index i_c_custkey;
create unique index i_c_custkey
on customer (c_custkey)
```

```
pctfree 0
initrans 10
tablespace ts_index
storage (freelists 84)
parallel
compute statistics
nologging;

host date
spool off;

host net stop oracleservicetpch
host net start oracleservicetpch
exit
!

#!/bin/ksh
sqlplus -s /NOLOG << !
connect / as sysdba
startup;
spool anlyz.out;
connect tpch/tpch;

host date

set timing on

execute dbms_stats.gather_schema_stats('TPCH' , estimate_percent => 1, degree => 32 , granularity => 'GLOBAL' );

connect / as sysdba
execute dbms_stats.gather_system_stats;
execute dbms_scheduler.disable('GATHER_STATS_JOB');
alter system switch logfile;

host date
spool off
exit
!
```

ixcre-analyz.sh

```
#!/bin/ksh
sqlplus -s /NOLOG << !
host date
connect tpch/tpch;

spool ixcre.out
host date
set echo on
set timing on

drop index i_l_orderkey;
create index i_l_orderkey
on lineitem (l_orderkey)
pctfree 5
initrans 10
tablespace ts_index
storage (freelist groups 4 freelists 84)
parallel
compute statistics
nologging;


drop index i_o_orderkey;
```

```
create unique index i_o_orderkey
on orders (o_orderkey)
pctfree 5
initrans 10
tablespace ts_index
storage (freelist groups 4 freelists 84 )
parallel
compute statistics
nologging;

drop index i_c_custkey;
create unique index i_c_custkey
on customer (c_custkey)
pctfree 0
initrans 10
tablespace ts_index
storage (freelists 84)
parallel
compute statistics
nologging;

host date
spool off;


spool anlyz.out;
connect tpch/tpch;

host date

set timing on

execute dbms_stats.gather_schema_stats('TPCH' , estimate_percent => 1, degree => 32 , granularity => 'GLOBAL' );

connect / as sysdba
execute dbms_stats.gather_system_stats;
execute dbms_scheduler.disable('GATHER_STATS_JOB');
alter system switch logfile;

host date
spool off
exit
!
```

APPENDIX C: Query Text & Output

-- using 717160520 as a seed to the RNG

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

-- Functional Query Definition
-- Approved February 1998

```
select
l_returnflag,
l_linestatus,
sum(l_quantity) as sum_qty,
sum(l_extendedprice) as sum_base_price,
sum(l_extendedprice * (1 - l_discount)) as sum_disc_price,
sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) as sum_charge,
avg(l_quantity) as avg_qty,
avg(l_extendedprice) as avg_price,
avg(l_discount) as avg_disc,
count(*) as count_order
from
lineitem
where
l_shipdate <= to_date ('1998-12-01','YYYY-MM-DD')  -  93
group by
l_returnflag,
l_linestatus
order by
l_returnflag,
l_linestatus
```

```
L_RETURNFLAG L_LINESTATUS SUM_QTY              SUM_BASE_PRICE
SUM_DISC_PRICE       SUM_CHARGE          AVG_QTY
AVG_PRICE            AVG_DISC            COUNT_ORDER
A            F           37795142625.00       56673501785799.90
53839798587628.00     55993369571208.80     25.50
38236.76             0.05                1482173068.00
N            F           986801952.00         1479667227558.55
1405678573470.13      1461921779364.50      25.50
38237.87             0.05                38696378.00
N            O           74276114603.00       111377358638761.00
105808524368123.00    110040903772321.00    25.50
38237.48             0.05                2912779709.00
R            F           37795090136.00       56673574737756.20
53839881846024.10     55993472963428.80     25.50
38237.23             0.05                1482156800.00


4 rows processed.
Query Processed in 1067.87 seconds.
```

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

-- Functional Query Definition
-- Approved February 1998

```
select * from (
select
s_acctbal,
s_name,
n_name,
p_partkey,
p_mfgr,
s_address,
s_phone,
```

```
s_comment
from
part,
supplier,
partsupp,
nation,
region
where
p_partkey = ps_partkey
and s_suppkey = ps_suppkey
and p_size = 40
and p_type like '%NICKEL'
and s_nationkey = n_nationkey
and n_regionkey = r_regionkey
and r_name = 'AFRICA'
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 = 'AFRICA'
)
order by
s_acctbal desc,
n_name,
s_name,
p_partkey
)
where rownum <= 100

S_ACCTBAL          S_NAME                N_NAME
P_PARTKEY          P_MFGR
S_ADDRESS                        S_PHONE
S_COMMENT
9999.99            Supplier#001002401    KENYA
31002400.00        Manufacturer#2
DvygUzOuiBASMn,hkVDYePt              24-524-206-4677
final deposits doubt slyly. special accounts doze fluffily. pending pa
9999.98            Supplier#006235057    KENYA
153735041.00       Manufacturer#2
fiw,mI9hSZ66bUz0oHgW,q               24-173-868-7564
carefully ironic deposits integrate quickly furiously even pin
9999.91            Supplier#006241432    MOZAMBIQUE
6241431.00         Manufacturer#3
qyhDTKayloZCI,Zr5pBd4et              26-432-206-9428
quickly regular courts nod slyly across the ironic requests. regular, regular requests pl
.
.
.
9997.97            Supplier#004608140    MOROCCO
182108121.00       Manufacturer#5
8D1ub7xeYU AmKYkcykuOfcvyfq59Ia      25-911-273-6153
carefully bold requests around the stealthy instructions will have to
9997.97            Supplier#003469379    MOZAMBIQUE
188469342.00       Manufacturer#4
wSUM8V,pjENEsgCJmvkgV 3RPu gAvuX     26-914-805-2798
slyly regular deposits engage furiously bold accounts-- furious
9997.96            Supplier#002152491    MOZAMBIQUE
9652490.00         Manufacturer#4
GWyu4NrwCMZ9vRY5M                    26-587-139-7763
final accounts against the regular theodolites use quickly quickly regular pinto b

100 rows processed.
```

Query Processed in 117.54 seconds.

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

-- Functional Query Definition
-- Approved February 1998

```
select * from (
select
l_orderkey,
sum(l_extendedprice * (1 - l_discount)) as revenue,
o_orderdate,
o_shippriority
from
customer,
orders,
lineitem
where
c_mktsegment = 'AUTOMOBILE'
and c_custkey = o_custkey
and l_orderkey = o_orderkey
and o_orderdate < to_date( '1995-03-01', 'YYYY-MM-DD')
and l_shipdate > to_date( '1995-03-01', 'YYYY-MM-DD')
group by
l_orderkey,
o_orderdate,
o_shippriority
order by
revenue desc,
o_orderdate)
where rownum <= 10
```

L_ORDERKEY	REVENUE	O_ORDERDATE	O_SHIPPRIORITY
3217511172.00	521252.94	20-FEB-95	0.00
4332612039.00	503063.78	22-FEB-95	0.00
702453537.00	502052.37	23-FEB-95	0.00
5450820225.00	501449.42	17-FEB-95	0.00
3712176198.00	500461.94	19-FEB-95	0.00
1647676162.00	498892.37	03-FEB-95	0.00
3764387078.00	497623.31	15-FEB-95	0.00
4692563456.00	496726.16	25-FEB-95	0.00
953223297.00	495049.18	24-FEB-95	0.00
691962789.00	494004.42	24-FEB-95	0.00

10 rows processed.
Query Processed in 79.15 seconds.

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

-- Functional Query Definition
-- Approved February 1998

```
select
o_orderpriority,
count(*) as order_count
from
orders
where
o_orderdate >= to_date( '1993-03-01',  'YYYY-MM-DD')
and o_orderdate < add_months(to_date( '1993-03-01', 'YYYY-MM-DD'),3)
and exists (
select
*
from
lineitem
where
l_orderkey = o_orderkey
and l_commitdate < l_receiptdate
```

```
)
group by
o_orderpriority
order by
o_orderpriority

O_ORDERPRIORITY ORDER_COUNT
1-URGENT        10532393.00
2-HIGH          10530784.00
3-MEDIUM        10528207.00
4-NOT SPECIFIED 10530318.00
5-LOW           10531299.00


5 rows processed.
Query Processed in 84.01 seconds.
```

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

```
-- Functional Query Definition
-- Approved February 1998


select
n_name,
sum(l_extendedprice * (1 - l_discount)) as revenue
from
customer,
orders,
lineitem,
supplier,
nation,
region
where
c_custkey = o_custkey
and l_orderkey = o_orderkey
and l_suppkey = s_suppkey
and c_nationkey = s_nationkey
and s_nationkey = n_nationkey
and n_regionkey = r_regionkey
and r_name = 'MIDDLE EAST'
and o_orderdate >= to_date( '1995-01-01',  'YYYY-MM-DD')
and o_orderdate < add_months(to_date( '1995-01-01',  'YYYY-MM-DD'),  12)
group by
n_name
order by
revenue desc

N_NAME                REVENUE
EGYPT                53071179134.29
SAUDI ARABIA          53009263956.83
IRAQ                 52985779525.08
IRAN                 52980985929.63
JORDAN                52864641082.17


5 rows processed.
Query Processed in 334.38 seconds.
```

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

```
-- Functional Query Definition
-- Approved February 1998


select
sum(l_extendedprice * l_discount) as revenue
from
lineitem
where
l_shipdate >= to_date( '1995-01-01',  'YYYY-MM-DD')
and l_shipdate < add_months(to_date( '1995-01-01',  'YYYY-MM-DD'),  12)
```

```
and l_discount between 0.04 - 0.01 and 0.04 + 0.01
and l_quantity < 25

REVENUE
89408524769.32


1 row processed.
Query Processed in 66.99 seconds.
```

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

```
-- Functional Query Definition
-- Approved February 1998


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

SUPP_NATION          CUST_NATION          L_YEAR
REVENUE
INDIA                MOZAMBIQUE           1995.00
52870171446.03
INDIA                MOZAMBIQUE           1996.00
53056792714.31
MOZAMBIQUE           INDIA                1995.00
52989371850.21
MOZAMBIQUE           INDIA                1996.00
53161908807.57


4 rows processed.
Query Processed in 420.69 seconds.
```

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

```
-- Variant A
-- Approved February 1998


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

O_YEAR              MKT_SHARE
1995.00             0.04
1996.00             0.04


2 rows processed.
Query Processed in 242.28 seconds.
```

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

```
-- Functional Query Definition
-- Approved February 1998


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

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

NATION                O_YEAR            SUM_PROFIT
ALGERIA               1998.00         30866399248.00
ALGERIA               1997.00         52752477397.95
ALGERIA               1996.00         52837208987.79
ALGERIA               1995.00         52750452594.80
ALGERIA               1994.00         52824567634.14
ALGERIA               1993.00         52646655006.69
ALGERIA               1992.00         52796197207.49
ARGENTINA               1998.00           30950366455.76
ARGENTINA               1997.00           52750441349.95
ARGENTINA               1996.00           52884116483.00
ARGENTINA               1995.00           52686780662.48
ARGENTINA               1994.00           52741376989.06
.
.
.
.
.
UNITED STATES           1998.00           30964613276.25
UNITED STATES           1997.00           52684678548.15
UNITED STATES           1996.00           52890544951.16
UNITED STATES           1995.00           52710763741.63
UNITED STATES           1994.00           52689708165.23
UNITED STATES           1993.00           52691538017.71
UNITED STATES           1992.00           52848397952.18
VIETNAM               1998.00         31089337566.08
VIETNAM               1997.00         52839561385.14
VIETNAM               1996.00         52885673622.13
VIETNAM               1995.00         52853859917.12
VIETNAM               1994.00         52880731992.45
VIETNAM               1993.00         52870384355.10
VIETNAM               1992.00         53013380496.64


175 rows processed.
Query Processed in 1268.33 seconds.
```

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

```
-- Functional Query Definition
-- Approved February 1998


select  *  from  (
select
c_custkey,
c_name,
sum(l_extendedprice * (1 - l_discount)) as revenue,
c_acctbal,
n_name,
c_address,
c_phone,
c_comment
from
customer,
```

```
orders,
lineitem,
nation
where
c_custkey = o_custkey
and l_orderkey = o_orderkey
and o_orderdate >= to_date ('1993-09-01',  'YYYY-MM-DD')
and o_orderdate < add_months(  to_date( '1993-09-01',  'YYYY-MM-DD'),  3)
and l_returnflag = 'R'
and c_nationkey = n_nationkey
group by
c_custkey,
c_name,
c_acctbal,
c_phone,
n_name,
c_address,
c_comment
order by
revenue desc)
where rownum <= 20

C_CUSTKEY            C_NAME                  REVENUE
C_ACCTBAL            N_NAME
C_ADDRESS                        C_PHONE
C_COMMENT
44299021.00          Customer#044299021      881812.99
2617.95              VIETNAM
vrHBG30AtZHgKv9Hv                    31-481-157-6886
blithely unusual accounts use car
104888827.00         Customer#104888827      794741.04
8525.15              MOZAMBIQUE
rl0nvwyToiDwDC4 Jd,3xMVIdYEvAlaW0sEPg    26-485-530-2775
silent accounts cajole slyly blithely ironic theodolites. furiously regular dolphins wake blithely. qui
64687684.00          Customer#064687684      775540.80
753.45               VIETNAM
HT3 tSwoTD6ssIJU4j7w1gBysTIRF1          31-521-426-9327
boldly silent courts wake furiously ar
124956751.00         Customer#124956751      747613.31
5988.42              KENYA
xOQgvYrFx,VG                         24-284-803-7133
even deposits are after the final, express ideas. regular packages sleep furiously despite the qu
36781277.00          Customer#036781277      740906.20
7127.94              JAPAN
UWC8pIytKCn                          22-826-549-3043
fluffily enticing frets affix carefully blithely express requests. blithely final
69034381.00          Customer#069034381      739791.36
2528.41              UNITED KINGDOM
5ggHLEMIvQsL6JrEZoLD0eCKWZ           33-210-413-6423
carefully final instructions nag against the furiously unusual ideas. fluffily ironic requests ar
84476444.00          Customer#084476444      736150.68
-600.52              MOROCCO
TGoUuVSB1LA,                         25-111-708-3562
regular grouches dazzle daringly accounts. so
9697189.00           Customer#009697189      734401.34
5920.02              INDIA
E94JeU8Z3z4NdOCO6lDVMHEZPYtDTLSrfbDEns   18-153-129-6863
furiously ironic foxes nag slyly furiously express deposits. carefully even deposits
3212203.00           Customer#003212203      734286.53
7105.51              JAPAN
7HjYypzxYhvohROFoNeM                 22-938-582-3878
slyly special foxes around the special accounts believe quickly even acco
60792997.00          Customer#060792997      726502.47
8937.83              SAUDI ARABIA
szezYK5V2JPkS YrrTXr                 30-791-321-1208
special, even foxes use carefully blithely ironic deposits. even excuses according to th
77778565.00          Customer#077778565      720845.49
3766.55              FRANCE
j8jzSYwAPmprICbNmNaElNUIfL5eyvm         16-134-448-6884
pending, final instructions wake slyly above the furiously final asymptotes. blithely regular
26748218.00          Customer#026748218      720666.55
621.82               MOZAMBIQUE
```

PE8IWAaqW9DbekgggboJjNlOeFR9ucAyHI2P 26-717-321-3996
patterns above the ironic, special dolphins wake blithely
126097540.00 Customer#126097540 720113.23
5869.93 UNITED KINGDOM
aW5AadeJRi0og wxCd8hYF2L 33-969-565-1338
quickly even deposits snooze blithely ideas. unusual ac
92676937.00 Customer#092676937 719753.98
2374.51 ROMANIA
Ywg ,,JWHa3QIVtVw4pBPFsL10ZHTpwwaEUuBs6E 29-219-975-5262
final pinto beans integrate after the regular ideas. pending, unusual ideas are carefully final
43656283.00 Customer#043656283 719254.61
3526.49 RUSSIA
MKfyMUb6lo3hpbcx0ao4OvbHzIc8 32-225-739-6472
express excuses across the bravely ironic packages promise slyly c
50367541.00 Customer#050367541 718540.86
9672.38 ETHIOPIA
gEx wiDm,nAUDfZYvDYXE 15-440-527-2023
regular requests about the furiously final excuses use according to the daringly regular foxes. furiously
105568942.00 Customer#105568942 716606.37
811.53 INDIA
,6ZWxJcrObYoCGI 18-501-124-2882
even accounts are blithely regular packages. slyly ironic
41001910.00 Customer#041001910 714417.31
7564.41 MOZAMBIQUE
7WBRDvTECs09ICMw1m,MMtzLMmerM5YWh RxUky 26-985-925-4491
regular requests boost blithely slyly unusual accounts. u
120834526.00 Customer#120834526 710843.62
8151.01 BRAZIL
O9fDFf4lzHgnaMMgatoA9UIhUIsTFr 12-844-518-6734
ironic requests dazzle above the accounts. carefully ironic deposits u
31245100.00 Customer#031245100 709270.65
7386.42 ETHIOPIA
Y7DbGaRYlVwL 15-912-953-3332
furiously unusual dependencies are carefully regular pinto beans. unusual ideas sleep blithely. furiously unusu

20 rows processed.
Query Processed in 189.56 seconds.

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

-- Functional Query Definition
-- Approved February 1998

```
select
ps_partkey,
sum(ps_supplycost * ps_availqty) as value
from
partsupp,
supplier,
nation
where
ps_suppkey = s_suppkey
and s_nationkey = n_nationkey
and n_name = 'JAPAN'
group by
ps_partkey having
sum(ps_supplycost * ps_availqty) > (
select
sum(ps_supplycost * ps_availqty) * 0.0000001000
from
partsupp,
supplier,
nation
where
ps_suppkey = s_suppkey
and s_nationkey = n_nationkey
and n_name = 'JAPAN'
)
order by
value desc
```

```
PS_PARTKEY          VALUE
173441498.00        26831279.18
193573417.00        26380970.71
29937229.00         25852299.09
116521710.00        25743354.82
.
.
.
72423993.00         7998178.80
141283726.00        7998178.48
30474008.00         7998177.36
51577630.00         7998177.00
125734329.00        7998176.00
147899811.00        7998175.99
157900135.00        7998175.07
119027603.00        7998174.54
72874498.00         7998174.53
23209318.00         7998174.20


940077 rows processed.
Query Processed in 172.76 seconds.
```

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

```
-- Functional Query Definition
-- Approved February 1998


select
        l_shipmode,
        sum(case
                when o_orderpriority = '1-URGENT'
                        or o_orderpriority = '2-HIGH'
                        then 1
                else 0
        end) as high_line_count,
        sum(case
                when o_orderpriority <> '1-URGENT'
                        and o_orderpriority <> '2-HIGH'
                        then 1
                else 0
        end) as low_line_count
from
        orders,
        lineitem
where
        o_orderkey = l_orderkey
        and l_shipmode in ('FOB', 'SHIP')
        and l_commitdate < l_receiptdate
        and l_shipdate < l_commitdate
and l_receiptdate >= to_date( '1993-01-01',  'YYYY-MM-DD')
and l_receiptdate < add_months(to_date ('1993-01-01',  'YYYY-MM-DD'),  12)
group by
        l_shipmode
order by
        l_shipmode

L_SHIPMODE HIGH_LINE_COUNT      LOW_LINE_COUNT
FOB     6235565.00          9355745.00
SHIP    6239181.00          9361184.00


2 rows processed.
Query Processed in 182.82 seconds.
```

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

```
-- Functional Query Definition
-- Approved February 1998


select
```

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

C_COUNT	CUSTDIST
0.00	50000003.00
10.00	10797881.00
11.00	9398334.00
9.00	9348839.00
20.00	7721518.00
19.00	7285130.00
21.00	7054120.00
8.00	6199149.00
12.00	6027847.00
18.00	5953187.00
22.00	5510275.00
17.00	4238901.00
23.00	3645837.00
7.00	3170934.00
13.00	2888555.00
16.00	2650378.00
24.00	2019311.00
15.00	1552329.00
14.00	1420525.00
6.00	1255675.00
25.00	918978.00
5.00	380657.00
26.00	337121.00
27.00	96709.00
4.00	86705.00
28.00	20932.00
3.00	14541.00
29.00	3361.00
2.00	1720.00
30.00	400.00
1.00	122.00
31.00	25.00
32.00	1.00

33 rows processed.
Query Processed in 1250.92 seconds.

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

```
-- Functional Query Definition
-- Approved February 1998


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

```
        lineitem,
        part
where
        l_partkey = p_partkey
        and l_shipdate >= date '1998-01-01'
        and l_shipdate < date '1998-01-01' + interval '1' month

PROMO_REVENUE
16.68


1 row processed.
Query Processed in 48.48 seconds.
```

-- TPC-H/TPC-R Top Supplier Query (Q15)

```
-- Variant A
-- Approved February 1998


with   revenue0
as (select
l_suppkey supplier_no,
sum(l_extendedprice * (1 - l_discount)) total_revenue
from
lineitem
where
                l_shipdate >= date '1993-07-01'
                and l_shipdate < date '1993-07-01' + interval '3' month
group by
l_suppkey)
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            S_NAME
S_ADDRESS                        S_PHONE        TOTAL_REVENUE
44957.00             Supplier#000044957
UafLgwqMvGsvAE YUxP                  16-824-818-1822 3456913.92


1 row processed.
Query Processed in 122.87 seconds.


-- @(#)1.sql            2.1.6.2
```

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

```
-- Functional Query Definition
-- Approved February 1998


select
p_brand,
p_type,
p_size,
```

```
count(distinct ps_suppkey) as supplier_cnt
from
partsupp,
part
where
p_partkey = ps_partkey
and p_brand <> 'Brand#22'
and p_type not like 'MEDIUM BRUSHED%'
and p_size in (19, 27, 9, 40, 20, 46, 7, 44)
and ps_suppkey not in (
select
s_suppkey
from
supplier
where
s_comment like '%Customer%Complaints%'
)
group by
p_brand,
p_type,
p_size
order by
supplier_cnt desc,
p_brand,
p_type,
p_size
```

```
P_BRAND    P_TYPE                    P_SIZE                SUPPLIER_CNT
Brand#21   ECONOMY BURNISHED NICKEL  7.00                  4820.00
Brand#34   MEDIUM POLISHED COPPER    44.00                 4785.00
Brand#32   SMALL ANODIZED NICKEL     40.00                 4772.00
Brand#55   STANDARD BRUSHED COPPER   7.00                  4744.00
Brand#14   SMALL BURNISHED COPPER    9.00                  4742.00
.
.
.
.
Brand#11   MEDIUM PLATED NICKEL      19.00                 3826.00
Brand#15   LARGE PLATED NICKEL       20.00                 3825.00
Brand#34   PROMO BRUSHED TIN         40.00                 3818.00
Brand#23   LARGE POLISHED BRASS      20.00                 3816.00
Brand#34   LARGE BRUSHED TIN         7.00                  3791.00
Brand#12   STANDARD BURNISHED NICKEL 9.00                  3786.00


27840 rows processed.
Query Processed in 227.92 seconds.
```

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

```
-- Functional Query Definition
-- Approved February 1998


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

AVG_YEARLY
307024111.48

1 row processed.
Query Processed in 372.68 seconds.

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

```
-- Function Query Definition
-- Approved February 1998


select * from (
select
c_name,
c_custkey,
o_orderkey,
o_orderdate,
o_totalprice,
sum(l_quantity)
from
orders,
customer,
lineitem
where
o_orderkey in (
select
l_orderkey
from
lineitem
group by
l_orderkey having
sum(l_quantity) > 313
)
and c_custkey = o_custkey
and o_orderkey = l_orderkey
group by
c_name,
c_custkey,
o_orderkey,
o_orderdate,
o_totalprice
order by
o_totalprice desc,
o_orderdate
)
where rownum <= 100
```

```
C_NAME                C_CUSTKEY           O_ORDERKEY          O_ORDERDATE
O_TOTALPRICE          SUM(L_QUANTITY)
Customer#101152111    101152111.00        1499040929.00       11-MAR-98
602901.81             321.00
Customer#102538214    102538214.00        3953307941.00       27-MAY-98
602901.81             321.00
Customer#107212873    107212873.00        1538557767.00       20-OCT-95
589263.48             322.00
.
.
.

Customer#048014495    48014495.00         475161217.00        15-OCT-94
553102.52             324.00
Customer#065091115    65091115.00         4716326021.00       04-SEP-93
553062.68             329.00
Customer#132724502    132724502.00        2939224930.00       09-FEB-98
552912.05             315.00
Customer#116730493    116730493.00        107481504.00        09-APR-92
552844.50             325.00
Customer#113904524    113904524.00        2561748516.00       29-SEP-96
```

```
552844.50            325.00


100 rows processed.
Query Processed in 2146.09 seconds.
```

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

```
-- Functional Query Definition
-- Approved February 1998


select
sum(l_extendedprice* (1 - l_discount)) as revenue
from
lineitem,
part
where
(
p_partkey = l_partkey
and p_brand = 'Brand#53'
and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
and l_quantity >= 2 and l_quantity <= 2 + 10
and p_size between 1 and 5
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
)
or
(
p_partkey = l_partkey
and p_brand = 'Brand#33'
and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED PACK')
and l_quantity >= 19 and l_quantity <= 19 + 10
and p_size between 1 and 10
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
)
or
(
p_partkey = l_partkey
and p_brand = 'Brand#22'
and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
and l_quantity >= 20 and l_quantity <= 20 + 10
and p_size between 1 and 15
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
)

REVENUE
3482790491.73


1 row processed.
Query Processed in 460.11 seconds.
```

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

```
-- Function Query Definition
-- Approved February 1998


select
s_name,
s_address
from
supplier,
nation
where
s_suppkey in (
select
ps_suppkey
from
```

```
partsupp
where
ps_partkey in (
select
p_partkey
from
part
where
p_name like 'medium%'
)
and ps_availqty > (
select
0.5 * sum(l_quantity)
from
lineitem
where
l_partkey = ps_partkey
and l_suppkey = ps_suppkey
and l_shipdate >= to_date ('1995-01-01', 'YYYY-MM-DD')
and l_shipdate < add_months( to_date ('1995-01-01', 'YYYY-MM-DD'), 12)
)
)
and s_nationkey = n_nationkey
and n_name = 'IRAQ'
order by
s_name
```

```
S_NAME                S_ADDRESS
Supplier#000000221      aU oCBZs0CUUTU
Supplier#000000252      xcaKgadrug
Supplier#000000304      b4rSMq4y hBDMJgcNp
Supplier#000000407      WliGC47Vto2nh7mj
Supplier#000000672      iu9d66fGNBYX
.
.
.
Supplier#009998636      ,RVfualXEBO
Supplier#009998644      bUnvh,kzW5hQpVpWMKlTXE
Supplier#009998701      ZK7C76f4 N6lnkt
Supplier#009998703      9A9uanIPAO pOPheuXA1
Supplier#009998845      M,QEjAZTyh
Supplier#009998991      NS8kEDUIACj5STxP2f9rN
Supplier#009998997      FImYMOjKnCg0Q8bjYinW5,uOyRV3eZhShCu
Supplier#009999492      KUwUP bnKGoMsLyEpEjxJr9tJKAboDcE
Supplier#009999827      qtQ1pB3VYL1e3ph09ks9MG3IFbApyu4CKGrDYS
Supplier#009999998      AIgjccyI9ma EkgiskKuKRdxsSCvYm15pWJBBORJ


107109 rows processed.
Query Processed in 156.26 seconds.
```

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

```
-- Functional Query Definition
-- Approved February 1998


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

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

S_NAME	NUMWAIT
Supplier#003349040	43.00
Supplier#003575297	43.00
Supplier#003672543	40.00
Supplier#006569073	40.00
Supplier#003809724	39.00
Supplier#008869753	39.00
Supplier#000308206	38.00
Supplier#004166136	38.00
Supplier#004887844	38.00
Supplier#005122719	38.00
.	
.	
.	
Supplier#006004259	33.00
Supplier#006420554	33.00
Supplier#006444204	33.00
Supplier#006535481	33.00
Supplier#006587616	33.00
Supplier#006636933	33.00

100 rows processed.
Query Processed in 1908.12 seconds.

-- TPC-H/TPC-R Global Sales Opportunity Query (Q22)

-- Functional Query Definition
-- Approved February 1998

```
select
cntrycode,
count(*) as numcust,
sum(c_acctbal) as totacctbal
from
(
select
substr(c_phone, 1, 2) as cntrycode,
c_acctbal
from
customer
where
substr(c_phone,1, 2) in
('21', '27', '11', '13', '29', '19', '12')
```

```
and c_acctbal > (
select
avg(c_acctbal)
from
customer
where
c_acctbal > 0.00
and substr(c_phone, 1, 2) in
('21', '27', '11', '13', '29', '19', '12')
)
and not exists (
select
*
from
orders
where
o_custkey = c_custkey
)
) custsale
group by
cntrycode
order by
cntrycode
```

CNTRYCODE	NUMCUST	TOTACCTBAL
11	907370.00	6807837875.67
12	910749.00	6832911572.97
13	910468.00	6826927596.02
19	907738.00	6806150832.13
21	908557.00	6813748423.93
27	907952.00	6810333214.26
29	909384.00	6819759711.92

7 rows processed.
Query Processed in 189.20 seconds.

APPENDIX D: Seed & Query Substitution

Parameters++

Substitution Parameters for Stream 00

++

-- using 717160520 as a seed to the RNG

14	1998-01-01										
2	40	NICKEL	AFRICA								
9	firebrick										
20	medium	1995-01-01		IRAQ							
6	1995-01-01		0.04	25							
17	Brand#23	WRAP CASE									
18	313										
8	MOZAMBIQUE		AFRICA	SMALL POLISHED STEEL							
21	JAPAN										
13	express	deposits									
3	AUTOMOBILE		1995-03-01								
22	21	27	11	13	29	19	12				
16	Brand#22	MEDIUM BRUSHED			19	27	9	40	20	46	7	44
4	1993-03-01										
11	JAPAN	0.0000001000									
15	1993-07-01										
1	93										
10	1993-09-01										
19	Brand#53	Brand#33	Brand#22	2	19	20					
5	MIDDLE EAST		1995-01-01								
7	INDIA	MOZAMBIQUE									
12	FOB	SHIP	1993-01-01								

++

Substitution Parameters for Stream 01

++

-- using 717160521 as a seed to the RNG

21	ETHIOPIA		
3	FURNITURE		1995-03-18
18	315		
5	AFRICA	1995-01-01	
11	ALGERIA	0.0000001000	

7 ALGERIA INDIA

6 1995-01-01 0.02 24

20 turquoise 1994-01-01 ARGENTINA

17 Brand#25 WRAP JAR

12 TRUCK FOB 1993-01-01

16 Brand#52 PROMO ANODIZED 45 42 30 12 21 10 34 39

15 1996-02-01

13 express packages

10 1994-06-01

2 27 TIN ASIA

8 INDIA ASIA SMALL BURNISHED COPPER

14 1993-04-01

19 Brand#55 Brand#21 Brand#21 7 20 27

9 cyan

22 10 18 16 26 29 27 19

1 101

4 1995-10-01

++

Substitution Parameters for Stream 02

++

-- using 717160522 as a seed to the RNG

6 1995-01-01 0.07 24

17 Brand#22 WRAP CAN

14 1993-07-01

16 Brand#42 SMALL PLATED 4 19 20 8 7 9 10 25

19 Brand#52 Brand#54 Brand#25 2 10 23

10 1993-03-01

9 chiffon

2 15 STEEL AFRICA

15 1993-11-01

8 ALGERIA AFRICA STANDARD BRUSHED COPPER

5 AMERICA 1995-01-01

22 19 13 32 15 11 18 27

12 RAIL FOB 1993-01-01

7 PERU ALGERIA

13 express packages

18 312

1 109

4 1993-07-01

20 grey 1997-01-01 MOROCCO

3 AUTOMOBILE 1995-03-03

11 JORDAN 0.0000001000

21 RUSSIA

++

Substitution Parameters for Stream 03

++

-- using 717160523 as a seed to the RNG

8 PERU AMERICA STANDARD PLATED COPPER

5 ASIA 1995-01-01

4 1996-01-01

6 1995-01-01 0.05 25

17 Brand#24 SM CASE

7 INDONESIA PERU

1 117

18 314

22 10 14 28 19 25 24 23

14 1993-10-01

9 blue

10 1994-01-01

15 1996-05-01

11 ARGENTINA 0.0000001000

20 royal 1995-01-01 ETHIOPIA

2 3 BRASS EUROPE

21 KENYA

19 Brand#14 Brand#32 Brand#14 8 11 20

13 express packages

16 Brand#22 LARGE POLISHED 5 3 29 31 9 38 23 8

12 AIR SHIP 1993-01-01

3 FURNITURE 1995-03-20

++

Substitution Parameters for Stream 04

++

-- using 717160524 as a seed to the RNG

5 MIDDLE EAST 1996-01-01

21 FRANCE

14 1994-02-01

19 Brand#12 Brand#25 Brand#13 3 12 27

15 1994-02-01

17 Brand#21 SM JAR

12 REG AIR SHIP 1994-01-01

6 1996-01-01 0.02 24

4 1993-10-01

9	aquamarine									
8	INDONESIA	ASIA	STANDARD ANODIZED COPPER							
16	Brand#52	STANDARD ANODIZED		5	11	29	41	37	23	50	7
11	KENYA	0.0000001000								
2	41	NICKEL	AFRICA							
10	1994-10-01									
18	312									
1	64									
13	express	packages								
7	ARGENTINA	INDONESIA								
22	30	22	29	17	13	26	28			
3	MACHINERY	1995-03-05								
20	cornsilk	1994-01-01	SAUDI ARABIA							

++

Substitution Parameters for Stream 05

++

-- using 717160525 as a seed to the RNG

21	UNITED KINGDOM										
15	1996-09-01										
4	1996-05-01										
6	1996-01-01	0.08	24								
7	CHINA	ARGENTINA									
16	Brand#42	MEDIUM BURNISHED		8	5	34	50	32	26	18	10
19	Brand#14	Brand#53	Brand#13	8	13	23					
18	313										
14	1994-05-01										
22	11	23	12	31	18	25	29				
11	BRAZIL	0.0000001000									
13	express	requests									
3	FURNITURE	1995-03-22									
1	72										
2	28	TIN	EUROPE								
5	AFRICA	1996-01-01									
8	ARGENTINA	AMERICA		PROMO BRUSHED TIN							
20	olive	1997-01-01	IRAN								
12	SHIP	AIR	1996-01-01								
17	Brand#22	SM CAN									
10	1993-07-01										
9	violet										

++

Substitution Parameters for Stream 06

++

-- using 717160526 as a seed to the RNG

10 1994-04-01

3 MACHINERY 1995-03-07

15 1994-05-01

13 express requests

6 1996-01-01 0.05 25

8 PERU AMERICA PROMO PLATED TIN

9 spring

7 IRAN PERU

4 1994-02-01

11 MOROCCO 0.0000001000

22 34 33 29 30 20 11 26

18 315

12 MAIL SHIP 1994-01-01

1 80

5 AMERICA 1996-01-01

16 Brand#22 ECONOMY POLISHED 41 14 16 12 23 18 24 22

2 16 COPPER AMERICA

14 1994-08-01

19 Brand#21 Brand#45 Brand#52 3 14 30

20 azure 1996-01-01 UNITED STATES

17 Brand#24 LG CASE

21 MOROCCO

++

Substitution Parameters for Stream 07

++

-- using 717160527 as a seed to the RNG

18 312

8 INDONESIA ASIA PROMO ANODIZED TIN

20 lawn 1994-01-01 KENYA

21 INDIA

2 4 BRASS EUROPE

4 1996-09-01

22 24 29 13 33 23 17 22

17 Brand#21 LG JAR

1 88

11 CANADA 0.0000001000

9 seashell

19 Brand#23 Brand#23 Brand#51 9 15 26

3 BUILDING 1995-03-24

13 special requests

5 ASIA 1996-01-01

7 BRAZIL INDONESIA

10 1995-01-01

16 Brand#52 STANDARD BRUSHED 46 23 48 25 45 7 11 27

6 1996-01-01 0.02 24

14 1994-11-01

15 1996-12-01

12 TRUCK REG AIR 1994-01-01

APPENDIX E: Database Settings/Acid Scripts

Init_acidtpch.ora

```
INSTANCE_TYPE= RDBMS
DB_CREATE_FILE_DEST= '+aciddg1'
#Uncomment during the database creation
#control_files= '+aciddg1'
db_name= acidtpch
undo_management= auto

#need to uncomment this while running the database in normal mode (most of the time)
control_files= '+aciddg1/tpch/CONTROLFILE/Current.256.1'

statistics_level                = BASIC
#statistics_level               = typical
audit_trail                     = FALSE
compatible                      = 10.1.0.2
db_block_checksum               = false
db_block_size                   = 8192
db_cache_size                   = 4g
aq_tm_processes                                 = 0
db_file_multiblock_read_count     = 128
db_files                        = 512
db_name                         = acidtpch
db_writer_processes             = 2
dml_locks                       = 40000
enqueue_resources               = 40000
global_names                    = FALSE
shared_pool_size                = 1G
large_pool_size                 = 1G
#log_buffer                     = 33554432
log_buffer                      = 4194304
log_checkpoints_to_alert        = true
max_dump_file_size              = unlimited
timed_statistics                = false
nls_date_format                 = YYYY-MM-DD
open_cursors                    = 600
optimizer_index_cost_adj        = 25
optimizer_mode                  = CHOOSE
optimizer_features_enable       = 10.1.0.1
parallel_adaptive_multi_user    = TRUE
parallel_execution_message_size   = 16384
parallel_max_servers            = 256
parallel_min_servers            = 256
pga_aggregate_target            = 24G
processes                       = 500
query_rewrite_enabled           = true
recovery_parallelism            = 8
sessions                        = 500
transactions                    = 10
undo_management                 = auto
undo_retention                  = 100000
remote_login_passwordfile                   = none



BACKGROUND_DUMP_DEST= C:\ORACLE\ADMIN\acidTPCH\BDUMP
CORE_DUMP_DEST= C:\ORACLE\ADMIN\acidTPCH\CDUMP
USER_DUMP_DEST= C:\ORACLE\ADMIN\acidTPCH\UDUMP
```

dbcreate_1g.sql

```
startup pfile= c:\oracle\ora10\database\initacidtpch.ora nomount;
create database
  controlfile reuse
  logfile '+aciddg1' size 128m reuse,
                  '+aciddg1' size 128m reuse
  datafile '+aciddg1' size 512m reuse
  sysaux datafile '+aciddg1' size 256m reuse
  undo tablespace ts_undo1
     datafile  '+aciddg1' size 128m reuse
  default temporary tablespace ts_temp
     tempfile '+aciddg1' size 768m reuse
     extent management local uniform size 10m
  maxdatafiles 4000
;


spool c:\acid_tpch\catlog.txt
@c:\oracle\ora10\rdbms\admin\catalog.sql;
@c:\oracle\ora10\rdbms\admin\catparr.sql;
@c:\oracle\ora10\rdbms\admin\catproc.sql;
connect system/manager
@c:\oracle\ora10\rdbms\admin\utlxplan.sql;
@c:\oracle\ora10\sqlplus\admin\pupbld.sql;
spool off

spool sctso.log
drop tablespace ts_default including contents;
create tablespace ts_default
datafile '+aciddg1' size 600m reuse
extent management local
autoallocate
;

drop tablespace ts_l1 including contents;
create tablespace ts_l1
datafile '+aciddg1' size 1024m reuse
extent management dictionary default storage (initial 2560k next 256k maxextents unlimited
pctincrease 0)
;


drop tablespace ts_o1 including contents;
create tablespace ts_o1
datafile '+aciddg1' size 256m reuse
extent management dictionary default storage (initial 640k next 64k maxextents unlimited
pctincrease 0)
;

drop tablespace ts_rest including contents;
create tablespace ts_rest
datafile '+aciddg1' size 128m reuse
extent management dictionary default storage (initial 8m next 800k maxextents unlimited
pctincrease 0)
;
drop tablespace ts_psupp including contents;
create tablespace ts_psupp
datafile '+aciddg1' size 192m reuse
extent management dictionary default storage (initial 28m next 2m maxextents unlimited
pctincrease 0)
;
drop tablespace ts_default including contents;
create tablespace ts_default
datafile '+aciddg1' size 192m reuse
extent management dictionary default storage (initial 28m next 2m maxextents unlimited
pctincrease 0)
;
drop tablespace ts_index including contents;
create tablespace ts_index
datafile '+aciddg1' size 684m reuse
extent management local
uniform size 10m
;
spool off
```

```
drop user acidtpch cascade;
grant DBA
to acidtpch identified by acidtpch;

connect acidtpch/acidtpch;
drop directory acid_data_dir;

create directory acid_data_dir as 'C:\tpch_acid\dbgen\';

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

connect acidtpch/acidtpch;
drop table o_et;
create table o_et(
   o_orderdate         date ,
   o_orderkey          number ,
   o_custkey           number ,
   o_orderpriority     char(15) ,
   o_shippriority      number ,
   o_clerk             char(15) ,
   o_orderstatus       char(1) ,
   o_totalprice        number ,
   o_comment           varchar(79)
)
organization external (
type ORACLE_LOADER
default directory acid_data_dir
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
               )
               location (
'orders.tbl'
))
reject limit unlimited;

connect acidtpch/acidtpch;
drop table ps_et;
create table ps_et(
   ps_partkey          number ,
   ps_suppkey          number ,
   ps_supplycost       number ,
   ps_availqty         number ,
   ps_comment          varchar(199)
)
organization external (
type ORACLE_LOADER
default directory acid_data_dir
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
               )
               location (
'partsupp.tbl'
))
reject limit unlimited;

connect acidtpch/acidtpch;
drop table p_et;
create table p_et(
   p_partkey           number ,
   p_type              varchar(25) ,
   p_size              number ,
   p_brand             char(10) ,
   p_name              varchar(55) ,
   p_container         char(10) ,
   p_mfgr              char(25) ,
   p_retailprice       number ,
   p_comment           varchar(23)
)
organization external (
type ORACLE_LOADER
default directory acid_data_dir
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
               )
               location (
'part.tbl'
))
reject limit unlimited;

connect acidtpch/acidtpch;
drop table c_et;
create table c_et(
   c_custkey           number ,
   c_mktsegment        char(10) ,
   c_nationkey         number ,
   c_name              varchar(25) ,
   c_address           varchar(40) ,
   c_phone             char(15) ,
   c_acctbal           number ,
   c_comment           varchar(117)
)
organization external (
type ORACLE_LOADER
default directory acid_data_dir
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
               )
               location (
'customer.tbl'))
reject limit unlimited;

connect acidtpch/acidtpch;
drop table s_et;
create table s_et(
   s_suppkey           number ,
```

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

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

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

connect acidtpch/acidtpch;
alter table l_et parallel;
alter table o_et parallel;
alter table ps_et parallel;
alter table p_et parallel;
alter table c_et parallel;
alter table s_et parallel;

alter user acidtpch default tablespace ts_default;
alter user acidtpch temporary tablespace ts_temp;
```

```
set timing on
set echo on

connect acidtpch/acidtpch;
drop table lineitem;
create table lineitem(
   l_shipdate           ,
   l_orderkey           NOT NULL,
   l_discount           NOT NULL,
   l_extendedprice      NOT NULL,
   l_suppkey            NOT NULL,
   l_quantity           NOT NULL,
   l_returnflag         ,
   l_partkey            NOT NULL,
   l_linestatus         ,
   l_tax                NOT NULL,
   l_commitdate         ,
   l_receiptdate        ,
   l_shipmode           ,
   l_linenumber         NOT NULL,
   l_shipinstruct       ,
   l_comment
)
pctfree 1
pctused 99
initrans 10
tablespace ts_l1
storage (initial 2560k next 256k freelists 99)
parallel
nologging
partition by range (l_shipdate)
subpartition by hash(l_partkey)
subpartitions 4
(
partition item1 values less than (to_date('1992-01-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item2 values less than (to_date('1992-02-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item3 values less than (to_date('1992-03-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item4 values less than (to_date('1992-04-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item5 values less than (to_date('1992-05-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item6 values less than (to_date('1992-06-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item7 values less than (to_date('1992-07-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item8 values less than (to_date('1992-08-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item9 values less than (to_date('1992-09-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item10 values less than (to_date('1992-10-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item11 values less than (to_date('1992-11-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item12 values less than (to_date('1992-12-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item13 values less than (to_date('1993-01-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item14 values less than (to_date('1993-02-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item15 values less than (to_date('1993-03-01','YYYY-MM-DD'))
store in (ts_l1)
,
```

```
partition item16 values less than (to_date('1993-04-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item17 values less than (to_date('1993-05-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item18 values less than (to_date('1993-06-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item19 values less than (to_date('1993-07-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item20 values less than (to_date('1993-08-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item21 values less than (to_date('1993-09-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item22 values less than (to_date('1993-10-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item23 values less than (to_date('1993-11-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item24 values less than (to_date('1993-12-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item25 values less than (to_date('1994-01-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item26 values less than (to_date('1994-02-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item27 values less than (to_date('1994-03-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item28 values less than (to_date('1994-04-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item29 values less than (to_date('1994-05-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item30 values less than (to_date('1994-06-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item31 values less than (to_date('1994-07-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item32 values less than (to_date('1994-08-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item33 values less than (to_date('1994-09-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item34 values less than (to_date('1994-10-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item35 values less than (to_date('1994-11-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item36 values less than (to_date('1994-12-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item37 values less than (to_date('1995-01-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item38 values less than (to_date('1995-02-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item39 values less than (to_date('1995-03-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item40 values less than (to_date('1995-04-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item41 values less than (to_date('1995-05-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item42 values less than (to_date('1995-06-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item43 values less than (to_date('1995-07-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item44 values less than (to_date('1995-08-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item45 values less than (to_date('1995-09-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item46 values less than (to_date('1995-10-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item47 values less than (to_date('1995-11-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item48 values less than (to_date('1995-12-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item49 values less than (to_date('1996-01-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item50 values less than (to_date('1996-02-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item51 values less than (to_date('1996-03-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item52 values less than (to_date('1996-04-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item53 values less than (to_date('1996-05-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item54 values less than (to_date('1996-06-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item55 values less than (to_date('1996-07-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item56 values less than (to_date('1996-08-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item57 values less than (to_date('1996-09-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item58 values less than (to_date('1996-10-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item59 values less than (to_date('1996-11-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item60 values less than (to_date('1996-12-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item61 values less than (to_date('1997-01-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item62 values less than (to_date('1997-02-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item63 values less than (to_date('1997-03-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item64 values less than (to_date('1997-04-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item65 values less than (to_date('1997-05-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item66 values less than (to_date('1997-06-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item67 values less than (to_date('1997-07-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item68 values less than (to_date('1997-08-01','YYYY-MM-DD'))
store in (ts_l1)
```

```
,
partition item69 values less than (to_date('1997-09-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item70 values less than (to_date('1997-10-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item71 values less than (to_date('1997-11-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item72 values less than (to_date('1997-12-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item73 values less than (to_date('1998-01-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item74 values less than (to_date('1998-02-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item75 values less than (to_date('1998-03-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item76 values less than (to_date('1998-04-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item77 values less than (to_date('1998-05-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item78 values less than (to_date('1998-06-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item79 values less than (to_date('1998-07-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item80 values less than (to_date('1998-08-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item81 values less than (to_date('1998-09-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item82 values less than (to_date('1998-10-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item83 values less than (to_date('1998-11-01','YYYY-MM-DD'))
store in (ts_l1)
,
partition item84 values less than (MAXVALUE)
store in (ts_l1))
as select
   l_shipdate          ,
   l_orderkey          ,
   l_discount          ,
   l_extendedprice     ,
   l_suppkey           ,
   l_quantity          ,
   l_returnflag        ,
   l_partkey           ,
   l_linestatus        ,
   l_tax               ,
   l_commitdate        ,
   l_receiptdate       ,
   l_shipmode          ,
   l_linenumber        ,
   l_shipinstruct      ,
   l_comment
from l_et;

host date

connect acidtpch/acidtpch;
set timing on
set echo on

host date

drop table orders;
create table orders(
   o_orderdate         ,
   o_orderkey          NOT NULL,
   o_custkey           NOT NULL,
   o_orderpriority     ,
   o_shippriority      ,
   o_clerk             ,
   o_orderstatus       ,
   o_totalprice        ,
   o_comment
)
pctfree 1
pctused 99
initrans 10
tablespace ts_o1
storage (initial 640k next 64k freelists 99)
parallel
nologging
partition by range (o_orderdate)
subpartition by hash(o_custkey)
subpartitions 4
(
partition ord1 values less than (to_date('1992-01-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord2 values less than (to_date('1992-02-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord3 values less than (to_date('1992-03-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord4 values less than (to_date('1992-04-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord5 values less than (to_date('1992-05-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord6 values less than (to_date('1992-06-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord7 values less than (to_date('1992-07-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord8 values less than (to_date('1992-08-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord9 values less than (to_date('1992-09-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord10 values less than (to_date('1992-10-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord11 values less than (to_date('1992-11-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord12 values less than (to_date('1992-12-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord13 values less than (to_date('1993-01-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord14 values less than (to_date('1993-02-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord15 values less than (to_date('1993-03-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord16 values less than (to_date('1993-04-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord17 values less than (to_date('1993-05-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord18 values less than (to_date('1993-06-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord19 values less than (to_date('1993-07-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord20 values less than (to_date('1993-08-01','YYYY-MM-DD'))
store in (ts_o1)
,
```

```
partition ord21 values less than (to_date('1993-09-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord22 values less than (to_date('1993-10-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord23 values less than (to_date('1993-11-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord24 values less than (to_date('1993-12-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord25 values less than (to_date('1994-01-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord26 values less than (to_date('1994-02-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord27 values less than (to_date('1994-03-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord28 values less than (to_date('1994-04-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord29 values less than (to_date('1994-05-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord30 values less than (to_date('1994-06-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord31 values less than (to_date('1994-07-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord32 values less than (to_date('1994-08-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord33 values less than (to_date('1994-09-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord34 values less than (to_date('1994-10-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord35 values less than (to_date('1994-11-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord36 values less than (to_date('1994-12-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord37 values less than (to_date('1995-01-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord38 values less than (to_date('1995-02-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord39 values less than (to_date('1995-03-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord40 values less than (to_date('1995-04-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord41 values less than (to_date('1995-05-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord42 values less than (to_date('1995-06-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord43 values less than (to_date('1995-07-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord44 values less than (to_date('1995-08-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord45 values less than (to_date('1995-09-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord46 values less than (to_date('1995-10-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord47 values less than (to_date('1995-11-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord48 values less than (to_date('1995-12-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord49 values less than (to_date('1996-01-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord50 values less than (to_date('1996-02-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord51 values less than (to_date('1996-03-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord52 values less than (to_date('1996-04-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord53 values less than (to_date('1996-05-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord54 values less than (to_date('1996-06-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord55 values less than (to_date('1996-07-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord56 values less than (to_date('1996-08-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord57 values less than (to_date('1996-09-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord58 values less than (to_date('1996-10-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord59 values less than (to_date('1996-11-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord60 values less than (to_date('1996-12-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord61 values less than (to_date('1997-01-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord62 values less than (to_date('1997-02-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord63 values less than (to_date('1997-03-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord64 values less than (to_date('1997-04-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord65 values less than (to_date('1997-05-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord66 values less than (to_date('1997-06-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord67 values less than (to_date('1997-07-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord68 values less than (to_date('1997-08-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord69 values less than (to_date('1997-09-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord70 values less than (to_date('1997-10-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord71 values less than (to_date('1997-11-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord72 values less than (to_date('1997-12-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord73 values less than (to_date('1998-01-01','YYYY-MM-DD'))
store in (ts_o1)
```

```
,
partition ord74 values less than (to_date('1998-02-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord75 values less than (to_date('1998-03-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord76 values less than (to_date('1998-04-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord77 values less than (to_date('1998-05-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord78 values less than (to_date('1998-06-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord79 values less than (to_date('1998-07-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord80 values less than (to_date('1998-08-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord81 values less than (to_date('1998-09-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord82 values less than (to_date('1998-10-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord83 values less than (to_date('1998-11-01','YYYY-MM-DD'))
store in (ts_o1)
,
partition ord84 values less than (MAXVALUE)
store in (ts_o1)
)
as select
   o_orderdate         ,
   o_orderkey          ,
   o_custkey           ,
   o_orderpriority     ,
   o_shippriority      ,
   o_clerk             ,
   o_orderstatus       ,
   o_totalprice        ,
   o_comment
from o_et;

connect acidtpch/acidtpch



host date
drop table partsupp;
create table partsupp(
   ps_partkey          NOT NULL,
   ps_suppkey          NOT NULL,
   ps_supplycost       NOT NULL,
   ps_availqty         ,
   ps_comment          ,
constraint pk_partkey_suppkey_1 primary key(ps_partkey, ps_suppkey)
)
organization index
partition by hash(ps_partkey)
partitions 4
storage (initial 28m next 2m)
parallel
nologging
pctthreshold 50
tablespace ts_psupp
as select
   ps_partkey          ,
   ps_suppkey          ,
   ps_supplycost       ,
   ps_availqty         ,
   ps_comment
from ps_et;

host date
rem drop table customer;
create table customer(
   c_custkey           NOT NULL,
   c_mktsegment        ,
   c_nationkey         ,
   c_name              ,
   c_address           ,
   c_phone             ,
   c_acctbal           ,
   c_comment
)
pctfree 0
pctused 99
parallel
nologging
storage (initial 7m next 700k)
partition by hash (c_custkey)
partitions 4
tablespace ts_rest
as select
   c_custkey           ,
   c_mktsegment        ,
   c_nationkey         ,
   c_name              ,
   c_address           ,
   c_phone             ,
   c_acctbal           ,
   c_comment
from c_et;


drop table part;

create table part(
   p_partkey           NOT NULL,
   p_type              ,
   p_size              ,
   p_brand             ,
   p_name              ,
   p_container         ,
   p_mfgr              ,
   p_retailprice       ,
   p_comment
)
pctfree 0
pctused 99
parallel
nologging
storage (initial 8m next 800k)
partition by hash (p_partkey)
partitions 4
tablespace ts_rest
as select
   p_partkey           ,
   p_type              ,
   p_size              ,
   p_brand             ,
   p_name              ,
   p_container         ,
   p_mfgr              ,
   p_retailprice       ,
   p_comment
from p_et;

drop table supplier;
create table supplier(
   s_suppkey           NOT NULL,
   s_nationkey         ,
   s_comment           ,
   s_name              ,
   s_address           ,
   s_phone             ,
   s_acctbal
)
pctfree 0
pctused 99
parallel
nologging
storage (initial 1m next 128k)
partition by hash (s_suppkey)
partitions 4
```

```
tablespace ts_rest
as select
   s_suppkey          ,
   s_nationkey        ,
   s_comment          ,
   s_name             ,
   s_address          ,
   s_phone            ,
   s_acctbal
from s_et;


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

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

connect acidtpch/acidtpch
set timing on
set echo on


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

connect acidtpch/acidtpch
host date
set echo on
set timing on
drop index i_l_orderkey;
create index i_l_orderkey
on lineitem (l_orderkey) global partition by hash(l_orderkey)
partitions 4
pctfree 5
initrans 10
tablespace ts_index
storage (freelist groups 2 freelists 99)
parallel
compute statistics
nologging;


host date
set echo on
drop index i_o_orderkey;
create unique index i_o_orderkey
on orders (o_orderkey) global partition by hash (o_orderkey)
partitions 4
pctfree 5
initrans 10
tablespace ts_index
storage (freelist groups 2 freelists 99 )
parallel
compute statistics
nologging;


host date
set echo on
set timing on
drop index i_c_custkey;
create unique index i_c_custkey
on customer (c_custkey)
pctfree 5
initrans 10
tablespace ts_index
storage (freelist groups 2 freelists 99)
parallel
compute statistics
nologging;


host date
set timing on
execute dbms_stats.gather_schema_stats('acidtpch' , estimate_percent => 1, degree => 4 ,
granularity => 'GLOBAL' );
host date
connect / as sysdba
execute dbms_stats.gather_system_stats;
execute dbms_scheduler.disable('GATHER_STATS_JOB');
alter system switch logfile;
```

Acid Scripts

Atranspl.h

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

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

  DESCRIPTION

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

*/
#ifndef ATRANSPL_H

#define ATRANSPL_H

#include <stdio.h>

#ifdef ORA_NT
 #include <windows.h>
 #include <sys\types.h>
 #include <time.h>
 #include <stdio.h>
 #include <stdlib.h>
#else /* ORA_NT */
 #include <stdlib.h>
 #include <string.h>
 #include <sys/param.h>
 #include <sys/types.h>
 #include <time.h>
 #include <errno.h>
 #include <math.h>
#endif /* ORA_NT */

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

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

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

```
extern int errno;

#ifndef NULL
#define NULL 0
#endif

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

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#ifndef ub1
#define ub1 unsigned char
#endif

#define UNAME_LEN 64
#define WRITE_BUF_LEN 1024

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

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

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

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

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

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

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

#define OCIsexec(svch,stmh,errh,iter) \
   if((status=OCIStmtExecute(svch,stmh,errh,iter,0,NULL,NULL,OCI_DEFAULT)) !=
OCI_SUCCESS) \
      sql_error(errh,status,1); \
   else \
      DISCARD 0

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

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

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

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

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

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

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

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

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

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

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

#endif /* ATRANSPL_H */
```

Atranspl.c

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

/*

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

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

  NOTES
    <other useful comments, qualifications, etc.>

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

```
*/

#ifdef ORA_NT
 #include <windows.h>
 #include <sys\types.h>
 #include <sys\stat.h>
 #include <time.h>
 #include <stdio.h>
 #include <stdlib.h>
 #include <fcntl.h>
 #include <io.h>
#else /* ORA_NT */
 #include <stdio.h>
 #include <stdlib.h>
 #include <sys/types.h>
 #include <sys/stat.h>
 #include <fcntl.h>
#endif /*ORA_NT */

#include "atranspl.h"

/* Declare error handling functions */

double gettime();
void sql_error();
void usage();
void ACIDinit();
void ACIDexit();
int atoi();
#ifdef ORA_NT
#define lrand48() ((long) rand() << 15 | rand()) /* rand is 32 bits on NT */
#define srand48(v) srand(v)
#define drand48() (((double)rand()/(double)RAND_MAX))
#else
void srand48();
long lrand48();
#endif /* ORA_NT */

/* declarations for ORDERS */

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

/* declarations for LINEITEM */

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

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

sb2 l_npricei;

/* other declarations */

int delta = 0;
double rprice;
double cost;

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

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

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

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

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

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

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

time_t curr_time;         /* Current Time                       */

/* OCI handles */

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

/* OCI bind handles */

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

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

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

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

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

char sqlstmt[1024];

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

void usage()
{

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

  fprintf(stderr,"    proc_no      :the process number within this ACID\n");
  fprintf(stderr,"    num_streams  :the total number of ACID transaction streams\n");
  fprintf(stderr,"    commit       :1 to commit transaction, abort otherwise\n\n");
```

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

}


void ACIDexit() {

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

}



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

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

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

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

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

  ACIDexit();

  exit(1);
}

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

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

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

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

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

  proc_no = atoi(argv[1]);

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

  num_streams = atoi(argv[2]);

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

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

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

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

  /* Process  optional parameters */

  argc -= 4;
  argv += 4;

  while(--argc) {
    ++argv;
    switch(argv[0][0]) {
    case 'u':
      strncpy((char *) lname, ++(argv[0]), UNAME_LEN);
      if (strchr((char *) lname, '/') == NULL) {
        fprintf(stderr, "Login name must be in the format of userid/passwd\n");
        usage();
        exit(-1);
      }
      break;
    case 'i':
      if ((infile = fopen(++(argv[0]), "r")) == NULL) {
        fprintf(stderr,"Cannot open input file %s\n", argv[0]);
        fprintf(stderr,"%s\n",strerror(errno));
        exit(-1);
      }
      BIS(flag, INFILE);
      break;
    case 'o':
#ifdef ORA_NT
      if ((outfile = open(++(argv[0]), (O_RDWR | O_CREAT), S_IREAD | S_IWRITE)) == -1)
#else
      if ((outfile = open(++(argv[0]), (O_RDWR | O_SYNC | O_CREAT), S_IRWXU)) == -1)
```

```
#endif
      {
       fprintf(stderr,"Cannot open output file %s\n", argv[0]);
       fprintf(stderr,"%s\n",strerror(errno));
       exit(-1);
      }
      BIS(flag, OUTFILE);
      break;
    case 'd':
#ifdef ORA_NT
      if ((logfile = open(++(argv[0]), (O_RDWR | O_CREAT), S_IREAD | S_IWRITE)) == -1)
#else
      if ((logfile = open(++(argv[0]), (O_RDWR | O_SYNC | O_CREAT), S_IRWXU)) == -1)
#endif
      {
       fprintf(stderr,"Cannot open durability success file %s\n", argv[0]);
       fprintf(stderr,"%s\n",strerror(errno));
       exit(-1);
      }
      BIS(flag, LOGFILE);
      break;
    case 'b':
      num_iter = atoi(++(argv[0]));
      break;
    case 't':
      trig = atoi(++(argv[0]));
      break;
    case 's':
      slp = atoi(++(argv[0]));
      break;
    default:
      fprintf(stderr, "Unknown argument %s\n", argv[0]);
      usage();
      break;
    }
  }

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

  /* Initialize the cursors etc. */

  (void) ACIDinit();

  /* sleep for some time (triggering) */

#ifdef ORA_NT
  Sleep(trig * 1000);
#else
  sleep(trig);
#endif

  /* start doing the ACID transactions */

  tr_start = gettime();

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

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

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

    /* Obtain l_key from l_key query */

    OCIsexec(tpcsvc,curi,errhp,1);

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

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

    /* Generate delta if necessary */

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

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

    curr_time = time(NULL);

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

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

    OCIsexec(tpcsvc,curr,errhp,1);

    curr_time = time(NULL);

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

    FPRTF1(outfile, "Sleep %d seconds before COMMIT/ROLLBACK...\n\n", slp);
#ifdef ORA_NT
    Sleep(slp * 1000);
#else
    sleep(slp);
#endif

    /* Shall we commit? */

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

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

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

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

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

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

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

      FPRTF(outfile, "AFTER TRANSACTION:\n");
```

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

 tr_end = gettime();

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

 /*  Disconnect from ORACLE. */

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

 ACIDexit();

 exit(0);
}


void ACIDinit()
{

 /* run random seed */

 srand48(getpid());

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

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

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

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

 /* get username and password */

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

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

 OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcsrv,0,OCI_ATTR_SERVER,errhp);
 OCIaset(tpcusr,OCI_HTYPE_SESSION,lname,strlen(lname),OCI_ATTR_USERNAME,
       errhp);
 OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI_ATTR_PASSWORD,
       errhp);

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

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

 /* Enable session parallel dml */

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

 /* Enable session parallel ddl */

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


 /* Make session serializable */

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

 /* Set optimizer_index_cost_adj = 25 */

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

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


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

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


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

#endif /* NOLKEY */

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

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

 /* bind variables */

 OCIbbname(curr,l_key_bp,errhp,":l_key",ADR(l_key),SIZ(l_key),SQLT_INT);
 OCIbbname(curr,o_key_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),SQLT_INT);
 OCIbbname(curr,delta_bp,errhp,":delta",ADR(delta),SIZ(delta),SQLT_INT);
 OCIbbname(curr,l_pkey_bp,errhp,":l_pkey",ADR(l_pkey),SIZ(l_pkey),SQLT_INT);
 OCIbbname(curr,l_skey_bp,errhp,":l_skey",ADR(l_skey),SIZ(l_skey),SQLT_INT);
 OCIbbname(curr,l_quan_bp,errhp,":l_quan",ADR(l_quan),SIZ(l_quan),SQLT_INT);
 OCIbbname(curr,l_newquan_bp,errhp,":l_newquan",ADR(l_newquan),
       SIZ(l_newquan),SQLT_INT);
```

```
 OCIbbname(curr,l_tax_bp,errhp,":l_tax",ADR(l_tax),SIZ(l_tax),SQLT_FLT);
 OCIbbname(curr,l_disc_bp,errhp,":l_disc",ADR(l_disc),SIZ(l_disc),SQLT_FLT);
 OCIbbname(curr,l_eprice_bp,errhp,":l_eprice",ADR(l_eprice),SIZ(l_eprice),
        SQLT_FLT);
 OCIbbname(curr,l_neweprice_bp,errhp,":l_neweprice",ADR(l_neweprice),
        SIZ(l_neweprice),SQLT_FLT);

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

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

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

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

 /* bind variables */

 OCIbbname(cure1,l_neweprice1_bp,errhp,":l_neweprice",ADR(l_neweprice),
        SIZ(l_neweprice),SQLT_FLT);
 OCIbbname(cure1,l_newquan1_bp,errhp,":l_newquan",ADR(l_newquan),
        SIZ(l_newquan),SQLT_INT);
 OCIbbname(cure1,o_key1_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),SQLT_INT);
 OCIbbname(cure1,l_key1_bp,errhp,":l_key",ADR(l_key),SIZ(l_key),SQLT_INT);

 OCIbbname(cure2,o_newtprice2_bp,errhp,":o_newtprice",ADR(o_newtprice),
        SIZ(o_newtprice),SQLT_FLT);
 OCIbbname(cure2,o_key2_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),SQLT_INT);

}
```

Atranspl.sql

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


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

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

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

   select o_totalprice
                into o_tprice
                from orders
                where o_orderkey = o_key;
```

select l_quantity, l_extendedprice, l_partkey, l_suppkey, l_tax, l_discount

```
      into l_quan, l_eprice, l_pkey, l_skey, l_tax, l_disc
      from lineitem
      where l_orderkey = o_key
      and   l_linenumber = l_key;

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

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

   update orders
      set o_totalprice = o_newtprice
      where o_orderkey = o_key;

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

   EXIT;

 EXCEPTION
  WHEN not_serializable THEN
    ROLLBACK;
 END;
```

```
 END LOOP;

END doatrans;
END;
/
exit;
```

Gettime.c

```
#ifdef RCSID
static char *RCSid =
  "$Header: gettime.c 15-jul-99.14:27:44 mpoess Exp $ ";
#endif /* RCSID */

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

/*

  NAME
    gettime.c

  DESCRIPTION
    get wall clock time.
    get cpu time.

  FUNCTIONS
    get wall clock time.
    get cpu time.

  NOTES
    Both routines return time in seconds as a double.
  MODIFIED   (MM/DD/YY)
  mpoess      07/15/99 - Creation
  mpoess      07/15/99 - Creation

*/


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

#ifdef ORA_NT
 #include <windows.h>
 #include <sys\types.h>
 #include <time.h>
 #include <stdio.h>
#else  /* ORA_NT */

#define SUN_OS5

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

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

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

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

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

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

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

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

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

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

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

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

# include <stdio.h>

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

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

#endif /* ORA_NT */

# ifdef __STDC__
double gettime(void)
# else
double gettime()
# endif /* __STDC__ */
{
#ifdef ORA_NT
  LARGE_INTEGER PFreq;        /* Ticks per Second */
  LARGE_INTEGER PCount;       /* Ticks Since 1970 */
  double count;
  double freq;

  if (!QueryPerformanceFrequency(&PFreq))
      return((clock_t)-1);
  if (!QueryPerformanceCounter(&PCount))
    return((clock_t)-1);
  freq = (double)PFreq.QuadPart;
  count = (double)PCount.QuadPart;
  return(count/freq);

#else /* ORA_NT */

#ifdef TIME_W_GETTIME
  struct timeval tv;

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

```
#endif /* TIME_W_GETTIME */

#ifdef TIME_W_TIMES
  struct tms buf;

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


#ifndef ORA_NT

double getcpu ()

{

#ifdef CPU_W_TIMES
  struct tms buf;

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

#ifdef CPU_W_GETRU
  struct rusage ru;
  double usecs;

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

}



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

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

{

#ifdef GETRU_STATS
  struct rusage ru;

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

#ifdef GET_P_STATS
  timeval_t tv;
  struct process_stats ru;

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

}



getru1 (kids)

int kids;

{

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

#ifdef GET_P_STATS
  timeval_t tv;

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

}



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

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

{

#ifdef GETRU_STATS
  struct rusage ru;

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

#ifdef GET_P_STATS
  timeval_t tv;
  struct process_stats ru;

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

}



#ifdef GETRU_STATS

print_ru (fp, ru)

FILE *fp;
struct rusage *ru;
```

```
{

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

}



diffru (ru2, ru)

struct rusage *ru2;
struct rusage *ru;

{

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

}

#endif /* GETRU_STATS */



#ifdef GET_P_STATS

print_ru (fp, ps)

FILE *fp;
struct process_stats *ps;

{

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

}



diffru (ru2, ru)

struct process_stats *ru2;
struct process_stats *ru;

{

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

}

#endif /* GET_P_STATS */

#endif /* !ORA_NT */
```

Gtime.c

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

/*

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

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

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

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

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

  NOTES
    <other useful comments, qualifications, etc.>

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

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

```
#include<stdlib.h>

#ifdef ORA_NT
#include <time.h>
double gettime();
#else
# include <sys/time.h>
#endif /* ORA_NT */

main ()
{

#ifdef ORA_NT
  printf("%lf", gettime());
#else
  struct timeval tv;

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

  printf ("%.2f\n", ((double) tv.tv_sec + (1.0e-6 * (double) tv.tv_usec)) ) ;
#endif
  exit(0);
}

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

Rankey.c

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

/*

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

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

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

#define ORDERCNT 150000.0

/* MK_SPARSE adopted from dss.h */

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

void sql_error();
void usage();
void ACIDinit();
long atol();
#ifdef ORA_NT
#define lrand48() ((long) rand() << 15 | rand()) /* rand is 32 bits on NT */
#define srand48(v) srand(v)
#define drand48() (((double)rand()/(double)RAND_MAX))
#else
void srand48();
long lrand48();
double drand48();
#endif /* ORA_NT */

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

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

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

/* OCI handles */

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

OCIBind *l_key_bp;
OCIBind *o_key_bp;

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

char sqlstmt[1024];


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


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

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

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

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

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

```
 ACIDexit();

 exit(1);
}


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

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

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

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

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

 argc -= 2;
 argv += 2;

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

 ACIDinit();

 /* initialize array for random numbers */

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

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

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

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

  /* Obtain l_key from l_key query */

  OCIsexec(tpcsvc,curi,errhp,1);

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

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

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

 ACIDexit();
 free(res);

}


void usage() {

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

void ACIDinit()
{

 /* run random seed */

 srand48(getpid());

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

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

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

 /* get username and password */

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

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

 OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcsrv,0,OCI_ATTR_SERVER,errhp);
 OCIaset(tpcusr,OCI_HTYPE_SESSION,lname,strlen(lname),OCI_ATTR_USERNAME,
       errhp);
 OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI_ATTR_PASSWORD,
       errhp);

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

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

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

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

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

Randpsup.c

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

/*

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

  DESCRIPTION
    Generate random keys for ACID PARTSUPP transactions:
    (Clause 4.2.3)
    PS_PARTKEY random within [SF*200000]
      and
```

```
    PS_SUPPKEY = (PS_PARTKEY + (i * ((S/4) + (int)(PS_PARTKEY - 1)
                /S))) % S + 1
    where i random within [0..3] and S = SF * 10000

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

*/

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

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

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

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

void usage();
double atof();
#ifdef ORA_NT
#define lrand48() ((long) rand() << 15 | rand())
#define srand48(v) srand(v)
#define drand48() (((double)rand()/(double)RAND_MAX))
#else
void srand48();
long lrand48();
double drand48();
#endif /* ORA_NT */


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

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

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

  /* seed the random number generator */

  srand48(getpid());

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

  PART_SUPP_BRIDGE(ps_skey, pkey, supp);

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

  exit(0);
}


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

A_query.sql

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

set serverout on;

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

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

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

exit;
```

A_query2.sql

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

set serverout on;

select
'BEFORE PARTSUPP QUERY' as STAGE,
```

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

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

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

exit;
```

D_hist.sql

```
Rem
Rem $Header: d_hist.sql 07-aug-99.21:33:08 mpoess Exp $
Rem
Rem d_hist.sql
Rem
Rem  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
Rem
Rem    NAME
Rem      d_hist.sql - <one-line expansion of the name>
Rem
Rem    DESCRIPTION
Rem      Creates a history table for ACID test purpose.
Rem
Rem    NOTES
Rem      <other useful comments, qualifications, etc.>
Rem
Rem    MODIFIED   (MM/DD/YY)
Rem    mpoess      08/07/99 - Creation
Rem    mpoess      08/07/99 - Created
Rem


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

drop table history;

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

exit;
```

Atom.sh

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

#. $KIT_DIR/env

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

usage() {

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

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

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

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

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

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

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

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

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

Consist.sh

```
#!sh
#
# $Header: consist.sh 08-aug-99.14:20:51 mpoess Exp $
#
# consist.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      consist.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#      Performs consistency tests.
#      Usage: consist.sh [-n iter] [-s number of stream] [-p prog]
#                        [-u usr/pswd] -h
#
```

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

#. $KIT_DIR/env

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

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

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

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

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

usage() {

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

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

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


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

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

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

i=0

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

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

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

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

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

i=0

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

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

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

wait

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

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

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

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

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

Consist.sql

```
Rem
Rem $Header: consist.sql 08-aug-99.16:59:17 mpoess Exp $
Rem
Rem consist.sql
Rem
Rem  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
Rem
Rem    NAME
Rem      consist.sql - <one-line expansion of the name>
Rem
Rem  DESCRIPTION
Rem       Verifies the consistency of TPC-D database using the
Rem       consistency condition.
Rem
Rem Usage:   sqlplus tpcd/tpcd @consist
Rem
Rem NOTE
Rem REQUIRES PACKAGES prvtotpt and dbmsotpt
rem
Rem    MODIFIED   (MM/DD/YY)
Rem    mpoess      08/08/99 - Creation
Rem    mpoess      08/08/99 - Created
Rem

set verify off
rem set termout on
rem set echo on


REM
REM Get today's date.
REM

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

set serverout on;

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


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

   diff := l_tprice - o_tprice;

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

END;
.
/


spool off
exit
```

Iso1.sh

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

#. $KIT_DIR/env

# May need to change the following:
RSH=rsh

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

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

USER=$DATABASE_USER
PROG=atranspl


rm -rf $TXN1FILE $TXN2FILE $KEYFILE

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

usage() {

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

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

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


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

# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

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

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

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

sleep 1

# start ACID transaction, Sleep for 60 second before COMMIT

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

# let's sleep 10 seconds before starting ACID query

sleep 15

# start ACID query with the same OKEY

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

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

Iso2.sh

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

#. $KIT_DIR/env

RSH=rsh

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

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

rm -rf $TXN1FILE $TXN2FILE $KEYFILE

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

usage() {

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

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

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


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

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

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

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

sleep 1

# start ACID transaction, Sleep for 30 second before ROLLBACK

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

# let's sleep 15 seconds before starting ACID query

sleep 15

# start ACID query with the same OKEY

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

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE
```

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

Iso3.sh

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

# . $KIT_DIR/env

# May need to change the following:
RSH=rsh

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

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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


# generate key files

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

sleep 1

# start ACID transaction, Sleep for 30 second before COMMIT

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

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

sleep 15

# start another ACID transaction with the same LKEY and OKEY
# but different DELTA

# Do not sleep before COMMIT so that we can see TXN2 has waited.

if [ "$RHOST" != "" ]
then
echo "Starting TXN2 on node $RHOST" >> $TXN2FILE
${RSH} -n ${RHOST} $PROG 2 2 1 1 i$KEYFILE u$USER s1 b1 >> $TXN2FILE &
else
$PROG 2 2 1 1 i$KEYFILE u$USER s1 b1 >> $TXN2FILE &
fi

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

Iso4.sh

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

# . $KIT_DIR/env

# May need to change the following:
RSH=rsh

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

DURA_DIR=$ACID_DIR/dura

TXN1FILE=$OUT_DIR/txn1$$.out
TXN2FILE=$OUT_DIR/txn2$$.out
```

```
KEYFILE=$OUT_DIR/key$$.out
ISOFILE=$OUT_DIR/iso4

USER=$DATABASE_USER
PROG=atranspl

rm -rf $TXN1FILE $TXN2FILE $KEYFILE

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

usage() {

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

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

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


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

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

sleep 1

# start ACID transaction, Sleep for 30 second before ROLLBACK

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

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

sleep 15

# start another ACID transaction with the same LKEY and OKEY
# but different DELTA

# Do not sleep before COMMIT so that we can see TXN2 has waited.

if [ "$RHOST" != "" ]
then
echo "Starting TXN2 on node $RHOST" >> $TXN2FILE
${RSH} -n ${RHOST} $PROG 2 2 1 1 i$KEYFILE u$USER s1 b1 >> $TXN2FILE &
else
$PROG 2 2 1 1 i$KEYFILE u$USER s1 b1 >> $TXN2FILE &
fi

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

Iso5.sh

```
#!sh
#
# $Header: iso5.sh 04-aug-99.09:21:45 mpoess Exp $
#
# iso5.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      iso5.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#      Usage: iso5.sh [-u user/password] [-n remote_node] -h
#      Options: See usage below
#  NOTES
#       For a cross node isolation test, assume the local node is
#       one of the participating nodes.  The other node can be
#       specified by the -n option.
#       You need to set the environment variable TPCD_KIT_DIR
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/04/99 - Creation
#    mpoess      08/04/99 - Creation
#

# . $KIT_DIR/env

# May need to change the following:
RSH=rsh

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

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

USER=$DATABASE_USER
PROG=atranspl

rm -rf $TXN1FILE $TXN2FILE $KEYFILE

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

usage() {

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

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

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


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

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

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

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

echo "Running ACID query BEFORE the start of Isolation Test 5" >> $TXN1FILE
```

```
echo "`date`" >> $TXN1FILE
echo "" >> $TXN1FILE
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >> $TXN1FILE
echo "" >> $TXN1FILE
echo "------------------------------------------------" >> $TXN1FILE

sleep 1

# start ACID transaction, Sleep for 60 second before COMMIT

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

# let's sleep 5 seconds before starting PARTSUPP query

sleep 5

# First generate PS_PARTKEY and PS_SUPPKEY

PSKEY=`randpsup 1`

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

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

wait

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

Iso6.sh

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

# . $KIT_DIR/env

# May need to change the following:
RSH=rsh

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

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

rm -rf $TXN1FILE $TXN2FILE $TXN3FILE $KEYFILE

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

usage() {

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

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

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


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

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

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

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

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

sleep 1

# start Query 1, use 0 as the delta

echo "Running Query 17b at `date`" >> $TXN1FILE
sqlplus $USER @q1 >> $TXN1FILE &

# sleep 2 seconds before starting ACID transaction

sleep 2

# start ACID transaction, COMMIT after one second

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

if [ "$RHOST" != "" ]
then
echo "Starting ACID transaction on node $RHOST" >> $TXN2FILE
```

```
${RSH} -n ${RHOST} $PROG 1 1 1 0 i$KEYFILE u$USER s1 >> $TXN2FILE &
else
$PROG 1 1 1 0 i$KEYFILE u$USER s1 >> $TXN2FILE &
fi

# start Query 17

sleep 2

echo "Running 2nd Query 17b at `date`" >> $TXN3FILE
sqlplus $USER @q1 >> $TXN3FILE &
# wait for everyone to finish

wait

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

cat $TXN1FILE $TXN2FILE $TXN3FILE >> $ISOFILE

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

Dura.sh

```
#!/bin/ksh
#
# $Header: dura.sh 08-aug-99.15:21:38 mpoess Exp $
#
# dura.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      dura.sh - <one-line expansion of the name>
#
#    DESCRIPTION
#      <short description of component this file declares/defines>
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/08/99 - Creation
#    mpoess      08/08/99 - Creation
#

. $KIT_DIR/env

# Create history table

# Count number of entries in the history table

SERVER="ultraperf2"

echo "----------"
echo "Capturing Process information before durability tests `date`"
rsh $SERVER -n -l spyda  ps -ef; date
echo "----------"

echo "----------"
echo "Starting the durability tests `date`"
run_acid.sh &
echo "----------"

sleep 1200

echo "----------"
echo "Collecting user information. `date`"
./cnt_user.sh pswong spyda ultraperf2 > dura/duraucnt 2>&1
echo "----------"

echo "----------"
echo "Capturing Process information while running Transactions `date`"
rsh $SERVER -n -l spyda  ps -ef; date
echo "----------"

echo "----------"
echo "Capturing disk information on Server: Ultraperf2 `date`"
rsh $SERVER -n -l spyda  vxprint -ht ; date
echo "----------"

echo "----------"
echo "Detaching mirror on data disk. `date`"
rsh $SERVER -n -l root  "vxplex -v ordr23 det ordr23-01"
echo "----------"

echo "----------"
echo "Capturing Disk information information on Server: Ultraperf2 `date`"
rsh $SERVER -n -l spyda  vxprint -ht ; date
echo "----------"

sleep 120

echo "----------"
echo "Capturing Process information after breaking data mirror. `date`"
rsh $SERVER -n -l spyda  ps -ef; date
echo "----------"


echo "----------"
echo "Detaching mirror on log2 disk. `date`"
rsh $SERVER -n -l root "vxplex -v log2 det log2-01"
echo "----------"

echo "----------"
echo "Capturing Disk information information on Server: Ultraperf2 `date`"
rsh $SERVER -n -l spyda  vxprint -ht ; date
echo "----------"

sleep 120

echo "----------"
echo "Capturing Process information after detaching log mirror. `date`"
rsh $SERVER -n -l spyda  ps -ef; date
echo "----------"

# Power Off
```

Ckpt.sh

```
#!sh
#
# $Header: ckpt.sh 08-aug-99.17:32:22 mpoess Exp $
#
# ckpt.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      ckpt.sh - <one-line expansion of the name>
#
#    DESCRIPTION
#      <short description of component this file declares/defines>
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/08/99 - Creation
#    mpoess      08/08/99 - Creation
#
#. $KIT_DIR/env

sqlplus -s /NOLOG << !

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

Count_tx.sql

```
#!/bin/ksh

STEM=$1
ITER=$2
OUT=$3
```

```
FIN=FALSE
while [ "$FIN" = "FALSE" ]
do
 s=0
 FIN=TRUE
 while [ $s -lt $STEM ]
 do
  nt=`grep "Transaction Completed" $OUT/dura${s} | wc -l`
  if [ $nt -lt $ITER ];then
    FIN=FALSE
  fi
  s=`expr $s + 1`
 done
 sleep 5
done
echo all streams have commited $ITER transactions
```

End_acid.sql

```
#!/bin/ksh
#
# $Header: end_acid.sh 08-aug-99.17:06:20 mpoess Exp $
#
# end_acid.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      end_acid.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#      end_cons.sh <pid of the durability run>
#      Options: See usage below
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/08/99 - Creation
#    mpoess      08/08/99 - Creation
#

. $KIT_DIR/env

OH=$ORACLE_HOME
# ACID_DIR=$OH/tpcd/audit set in env
OUT_DIR=$ACID_OUT/
#DURA_DIR=$ACID_OUT/dura
DURA_DIR=C:/tpch_acid/kit/acid_out/dura
RUN_ID_FILE=$ACID_DIR/run_id

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

# get history count

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

# perform the consistency

i=0
while [ $i -lt $STEM ]
do
  for j in `head -10 ${KEY}${i} | awk '{printf "%d ",$1}'`
  do
    sqlplus acidtpch/acidtpch @consist $j >> $DURA_DIR/duraconsa
  done
  i=`expr $i + 1`
done
i=0
while [ $i -lt $STEM ]
do
sample.sh $DURA${i} > ${DSMPL}${i} 2>&1
i=`expr $i + 1`
done
```

Sample.sh

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

# $1 durability output file

. $KIT_DIR/env

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



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

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

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

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

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

Sample.sql

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

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

exit;
```

Appendix F: Disk Configuration

Disk #	Type				Note:
0	Bas.	C: 32GB			
1	B	36GB	1TB db and Temp		Mount Pt. 1
2	B	36GB	1TB db and Temp		Mount Pt. 2
3	B	36GB	1TB db and Temp		Mount Pt. 3
4	B	36GB	1TB db and Temp		Mount Pt. 4
5	B	36GB	1TB db and Temp		Mount Pt. 5
6	B	36GB	1TB db and Temp		Mount Pt. 6
7	B	36GB	1TB db and Temp		Mount Pt. 7
8	B	36GB	1TB db and Temp		Mount Pt. 8
9	B	36GB	1TB db and Temp		Mount Pt. 9
10	Dyn.	36GB	U: UF's, Pagefile, Output		
11	B	36GB	1TB db and Temp		Mount Pt. 10
12	B	36GB	1TB db and Temp		Mount Pt. 11
13	B	36GB	1TB db and Temp		Mount Pt. 12
14	B	36GB	1TB db and Temp		Mount Pt. 13
15	B	36GB	1TB db and Temp		Mount Pt. 14
16	B	36GB	1TB db and Temp		Mount Pt. 15
17	B	36GB	1TB db and Temp		Mount Pt. 16
18	B	36GB	1TB db and Temp		Mount Pt. 17
19	B	36GB	1TB db and Temp		Mount Pt. 18
20	B	36GB	1TB db and Temp		Mount Pt. 19
21	B	36GB	1TB db and Temp		Mount Pt. 20
22	B	36GB	1TB db and Temp		Mount Pt. 21
23	B	36GB	1TB db and Temp		Mount Pt. 22
24	B	36GB	1TB db and Temp		Mount Pt. 23
25	Dyn.	36GB	U: UF's, Pagefile, Output		
26	B	36GB	1TB db and Temp		Mount Pt. 24
27	B	36GB	1TB db and Temp		Mount Pt. 25
28	B	36GB	1TB db and Temp		Mount Pt. 26
29	B	36GB	1TB db and Temp		Mount Pt. 27
30	B	36GB	1TB db and Temp		Mount Pt. 28
31	B	36GB	1TB db and Temp		Mount Pt. 29
32	B	36GB	1TB db and Temp		Mount Pt. 30
33	B	36GB	1TB db and Temp		Mount Pt. 31
34	B	36GB	1TB db and Temp		Mount Pt. 32
35	B	36GB	1TB db and Temp		Mount Pt. 33
36	B	36GB	1TB db and Temp		Mount Pt. 34
37	B	36GB	1TB db and Temp		Mount Pt. 35
38	B	36GB	1TB db and Temp		Mount Pt. 36
39	Dyn.	36GB	U: UF's, Pagefile, Output		

40	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 37
41	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 38
42	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 39
43	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 40
44	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 41
45	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 42
46	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 43
47	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 44
48	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 45
49	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 46
50	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 47
51	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 48
52	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 49
53	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 50
54	**Dyn.**	**36GB**	**U: UF's, Pagefile, Output**	
55	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 51
56	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 52
57	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 53
58	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 54
59	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 55
60	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 56
61	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 57
62	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 58
63	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 59
64	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 60
65	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 61
66	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 62
67	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 63
68	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 64
69	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 65
70	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 66
71	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 67
72	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 68
73	**Dyn.**	**36GB**	**U: UF's, Pagefile, Output**	
74	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 69
75	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 70
76	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 71
77	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 72
78	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 73
79	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 74
80	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 75
81	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 76
82	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 77
83	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 78
84	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 79
85	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 80
86	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 81

87	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 82
88	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 83
89	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 84
90	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 85
91	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 86
92	**Dyn.**	**36GB**	**U: UF's, Pagefile, Output**	
93	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 87
94	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 88
95	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 89
96	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 90
97	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 91
98	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 92
99	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 93
100	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 94
101	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 95
102	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 96
103	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 97
104	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 98
105	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 99
106	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 100
107	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 101
108	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 102
109	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 103
110	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 104
111	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 105
112	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 106
113	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 107
114	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 108
115	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 109
116	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 110
117	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 111
118	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 112
119	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 113
120	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 114
121	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 115
122	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 116
123	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 117
124	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 118
125	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 119
126	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 120
127	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 121
128	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 122
129	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 123
130	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 124
131	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 125
132	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 126
133	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 127

134	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 128
135	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 129
136	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 130
137	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 131
138	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 132
139	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 133
140	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 134
141	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 135
142	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 136
143	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 137
144	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 138
145	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 139
146	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 140
147	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 141
148	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 142
149	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 143
150	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 144
151	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 145
152	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 146
153	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 147
154	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 148
155	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 149
156	**B**	**36GB**	**ACID db**	
157	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 150
158	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 151
159	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 152
160	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 153
161	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 154
162	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 155
163	**B**	**36GB**	**1TB db** and **Temp**	Mount Pt. 156
164	**B**	**36GB**	**ACID db**	

ASM1TB.sql ----- Script to create ASM disk group.

```
Startup nomount
pfile=c:\oracle\admin\+data\pfile\init+data.ora
```

```
'c:\asm\disk1',
'c:\asm\disk2',
'c:\asm\disk3',
'c:\asm\disk4',
'c:\asm\disk5',
'c:\asm\disk6',
'c:\asm\disk7',
'c:\asm\disk8',
'c:\asm\disk9',
'c:\asm\disk10',
'c:\asm\disk11',
'c:\asm\disk12',
'c:\asm\disk13',
'c:\asm\disk14',
'c:\asm\disk15',
'c:\asm\disk16',
'c:\asm\disk17',
'c:\asm\disk18',
'c:\asm\disk19',
'c:\asm\disk20',
'c:\asm\disk21',
'c:\asm\disk22',
'c:\asm\disk23',
'c:\asm\disk24',
'c:\asm\disk25',
'c:\asm\disk26',
'c:\asm\disk27',
'c:\asm\disk28',
'c:\asm\disk29',
'c:\asm\disk30',
'c:\asm\disk31',
'c:\asm\disk32',
'c:\asm\disk33',
'c:\asm\disk34',
'c:\asm\disk35',
'c:\asm\disk36',
'c:\asm\disk37',
'c:\asm\disk38',
'c:\asm\disk39',
'c:\asm\disk40',
'c:\asm\disk41',
'c:\asm\disk42',
'c:\asm\disk43',
'c:\asm\disk44',
'c:\asm\disk45',
'c:\asm\disk46',
'c:\asm\disk47',
'c:\asm\disk48',
'c:\asm\disk49',
'c:\asm\disk50',
'c:\asm\disk51',
'c:\asm\disk52',
'c:\asm\disk53',
'c:\asm\disk54',
'c:\asm\disk55',
'c:\asm\disk56',
'c:\asm\disk57',
'c:\asm\disk58',
'c:\asm\disk59',
'c:\asm\disk60',
'c:\asm\disk61',
'c:\asm\disk62',
'c:\asm\disk63',
```

```
CREATE DISKGROUP dg1 normal REDUNDANCY disk
```

```
'c:\asm\disk64',
'c:\asm\disk65',
'c:\asm\disk66',
'c:\asm\disk67',
'c:\asm\disk68',
'c:\asm\disk69',
'c:\asm\disk70',
'c:\asm\disk71',
'c:\asm\disk72',
'c:\asm\disk73',
'c:\asm\disk74',
'c:\asm\disk75',
'c:\asm\disk76',
'c:\asm\disk77',
'c:\asm\disk78',
'c:\asm\disk79',
'c:\asm\disk80',
'c:\asm\disk81',
'c:\asm\disk82',
'c:\asm\disk83',
'c:\asm\disk84',
'c:\asm\disk85',
'c:\asm\disk86',
'c:\asm\disk87',
'c:\asm\disk88',
'c:\asm\disk89',
'c:\asm\disk90',
'c:\asm\disk91',
'c:\asm\disk92',
'c:\asm\disk93',
'c:\asm\disk94',
'c:\asm\disk95',
'c:\asm\disk96',
'c:\asm\disk97',
'c:\asm\disk98',
'c:\asm\disk99',
'c:\asm\disk100',
'c:\asm\disk101',
'c:\asm\disk102',
'c:\asm\disk103',
'c:\asm\disk104',
'c:\asm\disk105',
'c:\asm\disk106',
'c:\asm\disk107',
'c:\asm\disk108',
'c:\asm\disk109',
'c:\asm\disk110',
'c:\asm\disk111',
'c:\asm\disk112',
'c:\asm\disk113',
'c:\asm\disk114',
'c:\asm\disk115',
'c:\asm\disk116',
'c:\asm\disk117',
'c:\asm\disk118',
'c:\asm\disk119',
'c:\asm\disk120',
'c:\asm\disk121',
'c:\asm\disk122',
'c:\asm\disk123',
'c:\asm\disk124',
'c:\asm\disk125',
'c:\asm\disk126',
```

```
'c:\asm\disk127',
'c:\asm\disk128',
'c:\asm\disk129',
'c:\asm\disk130',
'c:\asm\disk131',
'c:\asm\disk132',
'c:\asm\disk133',
'c:\asm\disk134',
'c:\asm\disk135',
'c:\asm\disk136',
'c:\asm\disk137',
'c:\asm\disk138',
'c:\asm\disk139',
'c:\asm\disk140',
'c:\asm\disk141',
'c:\asm\disk142',
'c:\asm\disk143',
'c:\asm\disk144',
'c:\asm\disk145',
'c:\asm\disk146',
'c:\asm\disk147',
'c:\asm\disk148',
'c:\asm\disk149',
'c:\asm\disk150',
'c:\asm\disk151',
'c:\asm\disk152',
'c:\asm\disk153',
'c:\asm\disk154',
'c:\asm\disk155',
'c:\asm\disk156'
;
alter diskgroup dg1 alter template tempfile
attributes (unprotected);
```