
Compaq Computer Corporation

ProLiant 7000-6/400-M1
Oracle V8.0.4

TPC Benchmark™D
Full Disclosure Report

First Edition
June 1998

First Edition – June 26, 1998

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

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

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

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

© Copyright Compaq and Oracle Corporation 1998.

All rights reserved. Permission is hereby granted to reproduce this document in whole or in part provided the copyright notice printed above is set forth in full text or on the title page of each item reproduced.

Printed in the United States June 26, 1998

ProLiant and ProLiant 7000 are registered trademarks of Compaq Computer Corporation.

ORACLE, SQL*DBA, SQL*Loader, SQL*Net, SQL*Plus, Oracle8, Pro*C and PL/SQL are trademarks of Oracle Corporation.

TPC Benchmark, is a trademark of the Transaction Processing Performance Council.

All other brand or product names mentioned herein must be considered trademarks or registered trademarks of their respective owners.

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark™ D test conducted on the ProLiant 7000-6/400-M1 using Oracle8 Server™, in conformance with the requirements of the TPC Benchmark™ D Standard Specification, Revision 1.3.1. The operating system used for the benchmark was Microsoft Windows NT 4.0 (build 1381 – Service Pack 3). The application was written in C and compiled using Microsoft Visual C++ for Windows NT.

The TPC Benchmark™ D was developed by the Transaction Processing Performance Council (TPC). The TPC was founded to define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry.

Standard and Executive Summary Statements

Pages ii-iv contain the Executive Summary and Numerical Quantities Summary of the benchmark results for the Compaq ProLiant 7000-6/400-M1.

Auditor

The benchmark configuration, environment and methodology used to produce and validate the test results, and the pricing model used to calculate the cost per QppD and QthD were audited by Lorna Livingtree of Performance Metrics, Inc. to verify compliance with the relevant TPC specifications. The auditor's letter of attestation is attached as Appendix F.

COMPAQ COMPUTER CORPORATION		ProLiant 7000 - 6/400-M1		TPC-D 1.3.1	
				REPORT DATE: 26-June-98	
TOTAL SYSTEM COST		TPC-D POWER		TPC-D THROUGHPUT	
\$ 378,189		1694.6 QppD @ 100GB		553.1 QthD @ 100GB	
				\$ 390.65 \$/per QphD @ 100GB	
DATABASE SIZE	DATABASE MANAGER	OPERATING SYSTEM	OTHER SOFTWARE	AVAILABILITY DATE	
100GB	Oracle Version 8.0.4	Microsoft Windows NT Enterprise 4.0	MKS Toolkit Visual C++	15-July-98	
Query Time in Seconds Geometric Mean 207.9 Arithmetic Mean 591.1					
DATABASE LOAD TIME = 22 HOURS 21 MINUTES 6 SECONDS			DISK SIZE/DATABASE SIZE = 7.15		RAID: Y
SYSTEM COMPONENTS			DESCRIPTION		
NUMBER OF NODES			COMPAQ PROLIANT 7000-6/400-M1 (1 NODE)		
PROCESSORS			4 x 400MHZ PENTIUM II XEON®, 1M CACHE		
MEMORY			4 GB		
DISK DRIVES			84 x 9.1 GB AND 1 x 4.3 GB		
TOTAL GB OF STORAGE			715.4 GB		

Compaq Computer Corporation		ProLiant 7000- 6/400-M1		TPC-D REV 1.3.1		
				Report Date:	26-Jun-98	
Description	Part Number	Third Party	Unit Price	Qty	Extended Price	5 yr. Maint. Price
Server Hardware		Brand Pricing				
ProLiant 7000-6/400-M1	312150-001		25,800	1	25,800	
Pentium II Xeon /400MHz Processor						
Netelligent 10/100 UTP PCI NIC						
2 x Integrated Fast-SCSI-2 Controller						
CD-ROM.SmartStart,Configuration Utility						
ProLiant 7000 Maintenance Upgrade	276535-000		1,075	1		1,075
Enhanced Keyboard	110562-001		76	1	76	
42U 19" Rack	165753-001		1,712	1	1,712	
Rack Sidewall Kit (For 42U Rack)	165652-001		212	1	212	
ProLiant Xeon 400 - 1MB Processor Kit	312286-B21		4,945	3	14,835	
1 GB DIMM Kit	241774-B21		6,258	4	25,032	
Fibre Channel Host Adapter Kit / PCI	223180-B21		1,789	1	1,789	
Compaq VGA V50 - 15" Color Monitor	264150-001		268	1	268	
24 X 7 -4HR Server Maintenance Upgrade						11,308
Compaq Uninterruptible Power Supply T2000	242688-005		954	1	954	
Subtotal					70,678	12,383
Server Software						
Oracle 8.0.4 Enterprise Database Server for Windows NT		Oracle	50,160	1	50,160	66,640
MKS Toolkit for NT			499	1	499	5 Yr WTY
Microsoft Visual C++ and Subscription Service			499	1	499	2,495
Microsoft Windows NT Enterprise v. 4.0			3,999	1	3,999	
Subtotal					55,157	69,135
Storage Devices						
Fibre Channel Storage Hub 7	234453-001		1,113	1	1,113	
Fibre Channel Encl. & Array Cntrlr Kit, Rack	223100-001		6,883	6	41,297	
ProLiant Storage Maintenance Upgrade	276537-001		430	6		2,580
24 X 7 -4HR Storage Maintenance Upgrade						7,028
4.3 GB Pluggable SCSI-2 Drive	146742-005		751	1	751	Incl. Above
9.1 GB 10k rpm Pluggable SCSI-3 Drive	336357-B21		1,396	84	117,300	Incl. Above
4/8GB DAT Drive	295353-B21		766	1	766	
Subtotal					161,227	9,608
Total					\$287,062	\$91,127
Five-Year Cost of Ownership:					\$378,189	
Audited By: Lorna Livingtree 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 assumptions about past or future purchases are not permitted. All discounts reflect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark specifications. If you find that the stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank you.						

**Compaq Computer
Corporation****Numerical Quantities**

TPC-D Rev. 1.3.1

Report Date:
26-June-98**Measurement Results:**

Database Scaling (SF/Size)	=	100
Total Data Storage/Database Size	=	7.15
Database Load Time	=	22 hr. 21 min 6 sec
Query Streams for Throughput Test (S)	=	0
TPC-D Power Metric (QppD@100GB)	=	1694.6
TPC-D Throughput Metric (QthD@100GB)	=	553.1
Total System Price Over five years	=	\$378,189
TPC-D Power Throughput Metric (\$/QphD@100GB)	=	\$390.65

Measurement Intervals:

Measurement interval in Performance Test (TS)	=	11064.9 s
---	---	-----------

Duration of Stream Execution:

Stream ID	Seed	Start Date	Start Time	End Date	End Time	Total Time
Stream00	48480162	6/13/98	16:40:29	6/13/98	19:44:54	3:04:25
UF1		6/13/98	16:40:29	6/13/98	16:49:30	
UF2		6/13/98	19:37:01	6/13/98	19:44:54	

TPC-D Timing Intervals (in seconds):

Query	1	2	3	4	5	6	7	8	9
Stream 00	3398.0	44.8	652.4	526.9	1161.4	71.1	690.8	185.4	1867.8

Query	10	11	12	13	14	15	16	17	UF1	UF2
Stream 00	659.4	83.1	199.1	2.3	58.9	105.5	334.7	8.1	540.5	472.6

Table Of Contents

ABSTRACT	I
OVERVIEW	I
STANDARD AND EXECUTIVE SUMMARY STATEMENTS.....	I
AUDITOR	I
TABLE OF CONTENTS.....	V
GENERAL ITEMS.....	8
1.1 TEST SPONSOR	8
1.2 PARAMETER SETTINGS	8
1.3 CONFIGURATION ITEMS.....	9
CLAUSE 1: LOGICAL DATABASE DESIGN.....	11
2.1 TABLE DEFINITIONS.....	11
2.2 PHYSICAL ORGANIZATION OF DATABASE.....	11
2.3 HORIZONTAL PARTITIONING	11
2.4 REPLICATION	12
CLAUSE 2: QUERIES AND UPDATE FUNCTIONS RELATED ITEMS	13

3.1	QUERY LANGUAGE	13
3.2	RANDOM NUMBER GENERATION	13
3.3	SUBSTITUTION PARAMETERS GENERATION	13
3.4	QUERY TEXT AND OUTPUT DATA FROM DATABASE.....	14
3.5	QUERY SUBSTITUTION PARAMETERS AND SEEDS USED	14
3.6	UPDATE FUNCTION SOURCE CODE	14
3.7	DATABASE MAINTENANCE OPTION.....	15
CLAUSe 3: DATABASE SYSTEM PROPERTIES.....		16
4.1	ATOMICITY	16
4.2	CONSISTENCY	17
4.3	ISOLATION	17
4.4	DURABILITY	19
CLAUSe 4: SCALING AND DATA-BASE POPULATION		21
5.1	INITIAL CARDINALITY OF TABLES.....	21
5.2	DISTRIBUTION OF TABLES AND LOGS ACROSS MEDIA.....	22
5.3	PARTITIONING AND REPLICATION.....	37
5.4	DBGEN VERSION AND MODIFICATIONS	39
5.5	DATABASE CONTENT OF THE FIRST TEN ROWS	39
5.6	DATABASE LOAD TIME	40
5.7	DATA STORAGE RATIO.....	40
5.8	DATABASE LOAD MECHANISM DETAILS AND ILLUSTRATION.....	40
CLAUSe 5: PERFORMANCE METRICS AND EXECUTION RULES RELATED ITEMS		41
6.1	STEPS IN THE POWER TEST	41
6.2	TIMING INTERVALS	41
6.3	NUMBER OF STREAMS FOR THE THROUGHPUT TEST	42
6.4	START/FINISH TIME OF EACH QUERY STREAM	42
6.5	TOTAL ELAPSED TIME	42
6.6	START/FINISH TIME FOR UPDATE FUNCTION.....	42
6.7	TIMING INTERVALS FOR EACH QUERY AND EACH UPDATE	42
6.8	PERFORMANCE METRICS	43
6.9	REPRODUCIBILITY METHOD	43
CLAUSe 6: SUT AND DRIVER IMPLEMENTATION RELATED ITEMS		45
7.1	DRIVER	45
7.2	IMPLEMENTATION SPECIFIC LAYER (ISL)	46
CLAUSe 7: PRICING RELATED ITEMS		47
8.1	HARDWARE AND SOFTWARE USED IN THE PRICED SYSTEM	47

8.2 TOTAL FIVE YEAR PRICE	47
8.3 AVAILABILITY DATE	48
CLAUSe 8: RELATED ITEMS	49
9.1 AUDITOR’S REPORT & ATTESTATION LETTER.....	49
APPENDIX A: PARAMETER SETTINGS	51
APPENDIX B: TABLE DEFINITIONS	56
APPENDIX C: QUERY TEXT AND QUERY OUTPUT	122
APPENDIX D: SEED AND QUERY SUBSTITUTION PARAMETERS	156
APPENDIX E: IMPLEMENTATION SPECIFIC LAYER/SOURCE CODE	157
APPENDIX F: INITIAL TEN ROWS	181
APPENDIX G: DBGEN SOURCE CODE	188
APPENDIX H: PRICE QUOTE	203

General Items

1.1 Test Sponsor

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

Compaq Computer Corporation is the sponsor of this TPC Benchmark™ D.

1.2 Parameter Settings

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

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

- *Compiler optimization options*

This requirement can be satisfied by providing a full list of all parameters and options, as long as all those which 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 Appendix A.

1.3 Configuration Items

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

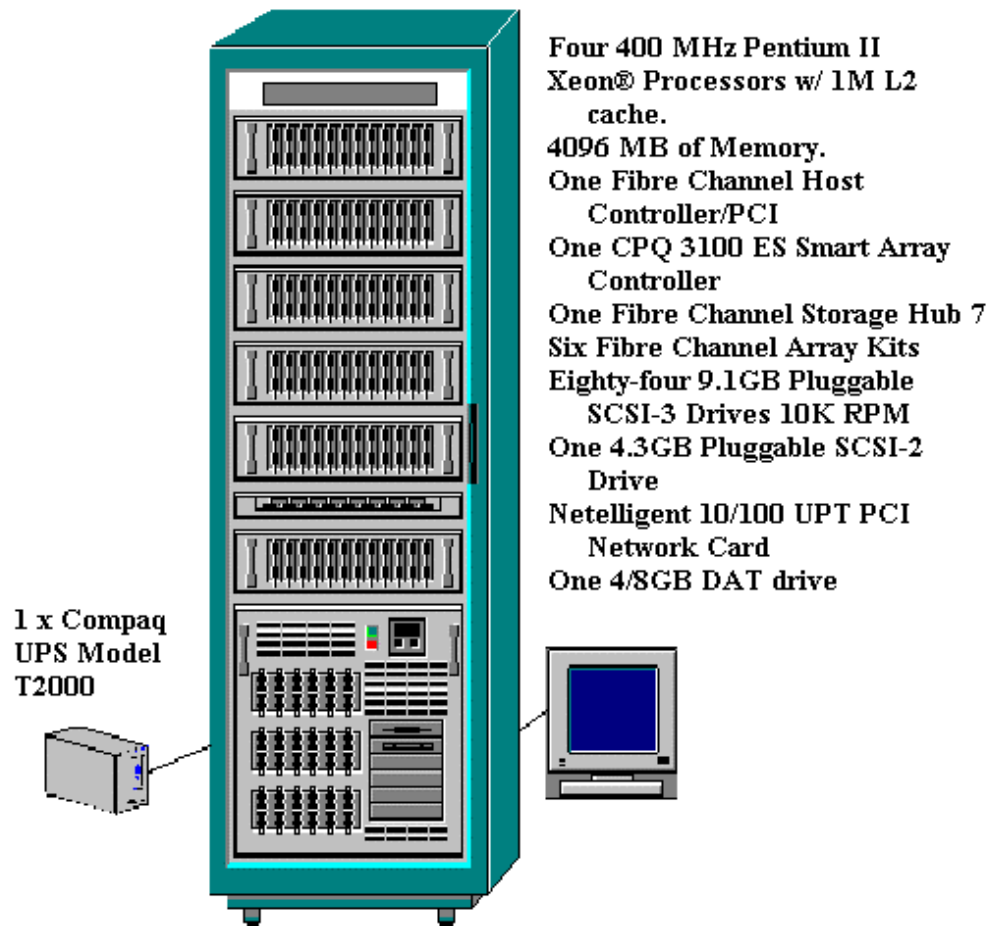
- *Number and type of processors*
- *Size of allocated memory, and any specific mapping/partitioning of memory unique to the test.*
- *Number and type of disk units (and controllers, if applicable).*
- *Number of channels or bus connections to disk units, including their protocol type.*
- *Number of LAN (e.g. Ethernet) Connections, including routers, workstations, terminals, etc., that were physically used in the test or are incorporated into the pricing structure.*
- *Type and the run-time execution location of software components (e.g., DBMS, query processing tools/languages, middle-ware components, software drivers, etc.).*

The server System Under Test (SUT), a Compaq ProLiant 7000-6/400-M1, depicted in Figure 1.1, consisted of:

- Four 400 MHz Pentium II Xeon® Processors w/ 1M L2 cache.
- 4096 MB of Memory.
- One Fibre Channel Host Controller/PCI
- One CPQ 3100 ES Smart Array Controller
- One Fibre Channel Storage Hub 7
- Six Fibre Channel Array Kits
- Eighty-four 9.1GB Pluggable SCSI-3 Drives 10K RPM
- One 4.3GB Pluggable SCSI-2 Drive
- Netelligent 10/100 UPT PCI Network Card
- One 4/8GB DAT drive

Figure 1. 1 Benchmarked and Priced Configuration

ProLiant 7000-6/400-1



Clause 1: Logical Database Design

2.1 Table Definitions

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

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

2.2 Physical Organization of Database

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.

2.3 Horizontal Partitioning

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

The database layout is described in the table in sections 5.2 and 5.3.

2.4 Replication

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

Replication was not used for this benchmark.

Clause 2: Queries and Update Functions related items

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.

A modified 1.2.0 version of DBGEN and version 1.2.0 of QGEN were used to generate the random numbers for this TPC-D benchmark. The modification to DBGEN was such that it would create and distribute the flat file data according to the way the database had them partitioned. The modified portions of DBGEN are supplied in Appendix G.

3.3 Substitution Parameters Generation

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

used, the version number, release number, modification number and patch level of QGEN must be disclosed.

The 1.2.0 version of QGEN was used to generate the substitution parameters.

3.4 Query Text and Output Data from 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

The minor query modifications used in this implementation include the following.

Wherever there is a date given in the WHERE clause of a select the TO_DATE function is used to convert the character input to the proper date format.

Wherever there is a date returned by the query the TO_CHAR function is used to convert the date format to the proper output format.

In some queries aliases have been assigned to the table names in the subqueries so that the columns are no longer ambiguous.

The Order table defined in the specification has been named the orders table due to conflict with a reserved word.

Whenever there is a time modification (i.e. add 3 months to a date) the ADD_MONTHS function is used to calculate and adjust the time delta and keep it in the date/time format.

CREATE TABLE statements have specific storage parameters in them. Namely, what tablespace to create them on.

3.5 Query Substitution Parameters and Seeds Used

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 Update Function Source Code

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

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

3.7 Database Maintenance Option

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

This implementation of the TPC-D benchmark uses the reset option.

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 prices 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. It was verified that the appropriate rows had been changed.

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, and were verified to have 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 prescribed number of ACID Transactions with randomly selected input parameters, and re-verify the consistency of the ORDER and LINEITEM tables.

1. The consistency of the ORDER and LINEITEM tables was verified based on a sample of O_ORDERKEYS.
2. 100 ACID transactions were submitted from each of 2 execution streams.
3. The consistency of the ORDER and LINEITEM tables was verified a second time with the same O_ORDERKEYS.

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 DELTA. The ACID transaction was suspended prior to rollback.
2. An ACID query was started for the same O_KEY used in step 1. The ACID query completed and did not see the uncommitted changes made by the ACID transaction.
3. The ACID transaction was rolled back.

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. A second ACID transaction, T2, was started using the same O_KEY and L_KEY and a different randomly selected DELTA.
3. T2 waited.
4. T1 was allowed to commit and then T2 completed.
5. It was verified that T2.L_EXTENDPRICE was calculated correctly.

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. A second ACID transaction, T2, was started using the same O_KEY and L_KEY and a different randomly selected DELTA.
3. T2 waited.
4. T1 was allowed to rollback and then T2 completed.
5. It was verified that T2.L_EXTENDPRICE was calculated correctly.

4.3.5 Concurrent Read/Write Transactions 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. The ACID transaction was suspended prior to commit.
2. A second ACID transaction, T2, was started that selected random values of PS_PARTKEY and PS_SUPPKEY and returned all columns of the PARTSUPP table for which PS_PARTKEY and PS_SUPPKEY were equal to the selected values.
3. T2 completes.
4. T1 was allowed to commit.
5. It was verified that the appropriate rows in the ORDER, LINEITEM, and HISTORY tables had been changed.

4.3.6 Update Transactions During Continuous Read-Only Query Streams

Demonstrate 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. T1 executes Query 1 against the qualification database where the substitution parameter is chosen from the interval [0..2159] so that the query runs for a sufficient length of time.
2. Before T1 completes, 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-D database tables or recovery log tables.

The disks containing the database logs and tables were RAID-1. The log loss, system crash and loss of data disks were combined into one

durability test. The auditor arbitrarily selected the log and data disks to be removed for the durability tests.

1. Two streams of ACID transactions were started.
2. After at least 200 ACID transactions had occurred and the streams were still running, one side of the mirrored set of logs was removed.
3. After it was determined that the test would still run with the loss of a log disk, (more ACID transactions had completed) a data disk attached to the Dazzler card was removed.
4. After it was determined that the test would still run with the loss of a log disk and a data disk attached to the Dazzler card (more ACID transactions had completed), a data disk attached to the fibre controller was removed.
5. After it was determined that the test would still run with the loss of a log disk and the loss of both data disks, the system was powered off by unplugging all power cables to the server.
6. We replaced the log disk and both data disks and allowed the controller to recover the mirrored pairs. The system was turned back on and the database ran through its recovery mode.
7. The counts in the success files, 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.

See section 4.4.1.

4.4.3 Memory Failure

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

See section 4.4.1.

Clause 4: Scaling and Database Population

5.1 Initial 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 5. 1: Initial number of Rows

Table	Occurrences
Orders	150,000,000
Lineitem	600,037,902
Customer	15,000,000
Part	20,000,000
Supplier	1,000,000
Partsupp	80,000,000
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....

The database tables are distributed across 6 Fibre Channel Array kits and one internal CPQ 3100 ES Smart Array Controller. The Smart 3100 ES contains one mirrored pair of disks where the Oracle8 log files are stored, one mirrored pair of disks for the system and rollback tablespaces and four mirrored pairs of disks containing Oracle8 data files. Six fibre channel array kits each have six mirrored pairs of disks containing Oracle8 data files. The flat files, for loading the database, were on an seventh fibre channel array kit, which contained eight 18GB drives on it but they were configured as one large logical drive. The operating system and the Oracle binaries were stored on a single non-mirrored 4.3GB disk drive on the embedded Fast/Wide SCSI-2 controller.

Table 5.2: Distribution of Tables Across Media

Array Kit	Oracle Link Name	Physical Device\Partition Number
Smart 3100 ES	ORA_LOG1	\device\harddisk0\partition1
Smart 3100 ES	ORA_LOG2	\device\harddisk0\partition2
Smart 3100 ES	ORA_SYS1	\device\harddisk1\partition1
Smart 3100 ES	ORA_ROLL1	\device\harddisk1\partition2
Smart 3100 ES	ORA_TEMP_1	\device\harddisk2\partition2
Smart 3100 ES	ORA_IND5_1	\device\harddisk2\partition1
Smart 3100 ES	ORA_IND1_1	\device\harddisk2\partition3
Smart 3100 ES	ORA_LINEITEM_1	\device\harddisk2\partition4
Smart 3100 ES	ORA_LINEITEM_41	\device\harddisk2\partition5
Smart 3100 ES	ORA_IND3_1	\device\harddisk2\partition6
Smart 3100 ES	ORA_PS_1	\device\harddisk2\partition7
Smart 3100 ES	ORA_IND4_1	\device\harddisk2\partition8
Smart 3100 ES	ORA_ORDER_1	\device\harddisk2\partition9
Smart 3100 ES	ORA_ORDER_41	\device\harddisk2\partition10
Smart 3100 ES	ORA_PARTS_1	\device\harddisk2\partition11
Smart 3100 ES	ORA_PARTS_41	\device\harddisk2\partition12
Smart 3100 ES	ORA_SUPPLIER_1	\device\harddisk2\partition13
Smart 3100 ES	ORA_CUSTOMER1_1	\device\harddisk2\partition14
Smart 3100 ES	ORA_CUSTOMER2_1	\device\harddisk2\partition15
Smart 3100 ES	ORA_CUSTOMER3_1	\device\harddisk2\partition16
Smart 3100 ES	ORA_CUSTOMER4_1	\device\harddisk2\partition17
Smart 3100 ES	ORA_CUSTOMER5_1	\device\harddisk2\partition18
Smart 3100 ES	ORA_TEMP_2	\device\harddisk3\partition2
Smart 3100 ES	ORA_IND5_2	\device\harddisk3\partition1
Smart 3100 ES	ORA_IND1_2	\device\harddisk3\partition3
Smart 3100 ES	ORA_LINEITEM_2	\device\harddisk3\partition4
Smart 3100 ES	ORA_LINEITEM_42	\device\harddisk3\partition5
Smart 3100 ES	ORA_IND3_2	\device\harddisk3\partition6
Smart 3100 ES	ORA_PS_2	\device\harddisk3\partition7

Smart 3100 ES	ORA_IND4_2	\device\harddisk3\partition8
Smart 3100 ES	ORA_ORDER_2	\device\harddisk3\partition9
Smart 3100 ES	ORA_ORDER_42	\device\harddisk3\partition10
Smart 3100 ES	ORA_PARTS_2	\device\harddisk3\partition11
Smart 3100 ES	ORA_PARTS_42	\device\harddisk3\partition12
Smart 3100 ES	ORA_SUPPLIER_2	\device\harddisk3\partition13
Smart 3100 ES	ORA_CUSTOMER1_2	\device\harddisk3\partition14
Smart 3100 ES	ORA_CUSTOMER2_2	\device\harddisk3\partition15
Smart 3100 ES	ORA_CUSTOMER3_2	\device\harddisk3\partition16
Smart 3100 ES	ORA_CUSTOMER4_2	\device\harddisk3\partition17
Smart 3100 ES	ORA_CUSTOMER5_2	\device\harddisk3\partition18
Smart 3100 ES	ORA_TEMP_3	\device\harddisk4\partition2
Smart 3100 ES	ORA_IND5_3	\device\harddisk4\partition1
Smart 3100 ES	ORA_IND1_3	\device\harddisk4\partition3
Smart 3100 ES	ORA_LINEITEM_3	\device\harddisk4\partition4
Smart 3100 ES	ORA_LINEITEM_43	\device\harddisk4\partition5
Smart 3100 ES	ORA_IND3_3	\device\harddisk4\partition6
Smart 3100 ES	ORA_PS_3	\device\harddisk4\partition7
Smart 3100 ES	ORA_IND4_3	\device\harddisk4\partition8
Smart 3100 ES	ORA_ORDER_3	\device\harddisk4\partition9
Smart 3100 ES	ORA_ORDER_43	\device\harddisk4\partition10
Smart 3100 ES	ORA_PARTS_3	\device\harddisk4\partition11
Smart 3100 ES	ORA_PARTS_43	\device\harddisk4\partition12
Smart 3100 ES	ORA_SUPPLIER_3	\device\harddisk4\partition13
Smart 3100 ES	ORA_CUSTOMER1_3	\device\harddisk4\partition14
Smart 3100 ES	ORA_CUSTOMER2_3	\device\harddisk4\partition15
Smart 3100 ES	ORA_CUSTOMER3_3	\device\harddisk4\partition16
Smart 3100 ES	ORA_CUSTOMER4_3	\device\harddisk4\partition17
Smart 3100 ES	ORA_CUSTOMER5_3	\device\harddisk4\partition18
Smart 3100 ES	ORA_TEMP_4	\device\harddisk5\partition2
Smart 3100 ES	ORA_IND5_4	\device\harddisk5\partition1
Smart 3100 ES	ORA_IND1_4	\device\harddisk5\partition3
Smart 3100 ES	ORA_LINEITEM_4	\device\harddisk5\partition4
Smart 3100 ES	ORA_LINEITEM_44	\device\harddisk5\partition5
Smart 3100 ES	ORA_IND3_4	\device\harddisk5\partition6
Smart 3100 ES	ORA_PS_4	\device\harddisk5\partition7
Smart 3100 ES	ORA_IND4_4	\device\harddisk5\partition8
Smart 3100 ES	ORA_ORDER_4	\device\harddisk5\partition9
Smart 3100 ES	ORA_ORDER_44	\device\harddisk5\partition10
Smart 3100 ES	ORA_PARTS_4	\device\harddisk5\partition11
Smart 3100 ES	ORA_PARTS_44	\device\harddisk5\partition12
Smart 3100 ES	ORA_SUPPLIER_4	\device\harddisk5\partition13
Smart 3100 ES	ORA_CUSTOMER1_4	\device\harddisk5\partition14
Smart 3100 ES	ORA_CUSTOMER2_4	\device\harddisk5\partition15
Smart 3100 ES	ORA_CUSTOMER3_4	\device\harddisk5\partition16
Smart 3100 ES	ORA_CUSTOMER4_4	\device\harddisk5\partition17
Smart 3100 ES	ORA_CUSTOMER5_4	\device\harddisk5\partition18
Fibre - 1	ORA_TEMP_5	\device\harddisk7\partition2
Fibre - 1	ORA_IND5_5	\device\harddisk7\partition1

Fibre - 1	ORA_IND1_5	\device\harddisk7\partition3
Fibre - 1	ORA_LINEITEM_5	\device\harddisk7\partition4
Fibre - 1	ORA_LINEITEM_45	\device\harddisk7\partition5
Fibre - 1	ORA_IND3_5	\device\harddisk7\partition6
Fibre - 1	ORA_PS_5	\device\harddisk7\partition7
Fibre - 1	ORA_IND4_5	\device\harddisk7\partition8
Fibre - 1	ORA_ORDER_5	\device\harddisk7\partition9
Fibre - 1	ORA_ORDER_45	\device\harddisk7\partition10
Fibre - 1	ORA_PARTS_5	\device\harddisk7\partition11
Fibre - 1	ORA_PARTS_45	\device\harddisk7\partition12
Fibre - 1	ORA_SUPPLIER_5	\device\harddisk7\partition13
Fibre - 1	ORA_CUSTOMER1_5	\device\harddisk7\partition14
Fibre - 1	ORA_CUSTOMER2_5	\device\harddisk7\partition15
Fibre - 1	ORA_CUSTOMER3_5	\device\harddisk7\partition16
Fibre - 1	ORA_CUSTOMER4_5	\device\harddisk7\partition17
Fibre - 1	ORA_CUSTOMER5_5	\device\harddisk7\partition18
Fibre - 1	ORA_TEMP_6	\device\harddisk8\partition2
Fibre - 1	ORA_IND5_6	\device\harddisk8\partition1
Fibre - 1	ORA_IND1_6	\device\harddisk8\partition3
Fibre - 1	ORA_LINEITEM_6	\device\harddisk8\partition4
Fibre - 1	ORA_LINEITEM_46	\device\harddisk8\partition5
Fibre - 1	ORA_IND3_6	\device\harddisk8\partition6
Fibre - 1	ORA_PS_6	\device\harddisk8\partition7
Fibre - 1	ORA_IND4_6	\device\harddisk8\partition8
Fibre - 1	ORA_ORDER_6	\device\harddisk8\partition9
Fibre - 1	ORA_ORDER_46	\device\harddisk8\partition10
Fibre - 1	ORA_PARTS_6	\device\harddisk8\partition11
Fibre - 1	ORA_PARTS_46	\device\harddisk8\partition12
Fibre - 1	ORA_SUPPLIER_6	\device\harddisk8\partition13
Fibre - 1	ORA_CUSTOMER1_6	\device\harddisk8\partition14
Fibre - 1	ORA_CUSTOMER2_6	\device\harddisk8\partition15
Fibre - 1	ORA_CUSTOMER3_6	\device\harddisk8\partition16
Fibre - 1	ORA_CUSTOMER4_6	\device\harddisk8\partition17
Fibre - 1	ORA_CUSTOMER5_6	\device\harddisk8\partition18
Fibre - 1	ORA_TEMP_7	\device\harddisk9\partition2
Fibre - 1	ORA_IND5_7	\device\harddisk9\partition1
Fibre - 1	ORA_IND1_7	\device\harddisk9\partition3
Fibre - 1	ORA_LINEITEM_7	\device\harddisk9\partition4
Fibre - 1	ORA_LINEITEM_47	\device\harddisk9\partition5
Fibre - 1	ORA_IND3_7	\device\harddisk9\partition6
Fibre - 1	ORA_PS_7	\device\harddisk9\partition7
Fibre - 1	ORA_IND4_7	\device\harddisk9\partition8
Fibre - 1	ORA_ORDER_7	\device\harddisk9\partition9
Fibre - 1	ORA_ORDER_47	\device\harddisk9\partition10
Fibre - 1	ORA_PARTS_7	\device\harddisk9\partition11
Fibre - 1	ORA_PARTS_47	\device\harddisk9\partition12
Fibre - 1	ORA_SUPPLIER_7	\device\harddisk9\partition13
Fibre - 1	ORA_CUSTOMER1_7	\device\harddisk9\partition14
Fibre - 1	ORA_CUSTOMER2_7	\device\harddisk9\partition15

Fibre - 1	ORA_CUSTOMER3_7	\\device\\harddisk9\\partition16
Fibre - 1	ORA_CUSTOMER4_7	\\device\\harddisk9\\partition17
Fibre - 1	ORA_CUSTOMER5_7	\\device\\harddisk9\\partition18
Fibre - 1	ORA_TEMP_8	\\device\\harddisk10\\partition2
Fibre - 1	ORA_IND5_8	\\device\\harddisk10\\partition1
Fibre - 1	ORA_IND1_8	\\device\\harddisk10\\partition3
Fibre - 1	ORA_LINEITEM_8	\\device\\harddisk10\\partition4
Fibre - 1	ORA_LINEITEM_48	\\device\\harddisk10\\partition5
Fibre - 1	ORA_IND3_8	\\device\\harddisk10\\partition6
Fibre - 1	ORA_PS_8	\\device\\harddisk10\\partition7
Fibre - 1	ORA_IND4_8	\\device\\harddisk10\\partition8
Fibre - 1	ORA_ORDER_8	\\device\\harddisk10\\partition9
Fibre - 1	ORA_ORDER_48	\\device\\harddisk10\\partition10
Fibre - 1	ORA_PARTS_8	\\device\\harddisk10\\partition11
Fibre - 1	ORA_PARTS_48	\\device\\harddisk10\\partition12
Fibre - 1	ORA_SUPPLIER_8	\\device\\harddisk10\\partition13
Fibre - 1	ORA_CUSTOMER1_8	\\device\\harddisk10\\partition14
Fibre - 1	ORA_CUSTOMER2_8	\\device\\harddisk10\\partition15
Fibre - 1	ORA_CUSTOMER3_8	\\device\\harddisk10\\partition16
Fibre - 1	ORA_CUSTOMER4_8	\\device\\harddisk10\\partition17
Fibre - 1	ORA_CUSTOMER5_8	\\device\\harddisk10\\partition18
Fibre - 1	ORA_TEMP_9	\\device\\harddisk11\\partition2
Fibre - 1	ORA_IND5_9	\\device\\harddisk11\\partition1
Fibre - 1	ORA_IND1_9	\\device\\harddisk11\\partition3
Fibre - 1	ORA_LINEITEM_9	\\device\\harddisk11\\partition4
Fibre - 1	ORA_LINEITEM_49	\\device\\harddisk11\\partition5
Fibre - 1	ORA_IND3_9	\\device\\harddisk11\\partition6
Fibre - 1	ORA_PS_9	\\device\\harddisk11\\partition7
Fibre - 1	ORA_IND4_9	\\device\\harddisk11\\partition8
Fibre - 1	ORA_ORDER_9	\\device\\harddisk11\\partition9
Fibre - 1	ORA_ORDER_49	\\device\\harddisk11\\partition10
Fibre - 1	ORA_PARTS_9	\\device\\harddisk11\\partition11
Fibre - 1	ORA_PARTS_49	\\device\\harddisk11\\partition12
Fibre - 1	ORA_SUPPLIER_9	\\device\\harddisk11\\partition13
Fibre - 1	ORA_CUSTOMER1_9	\\device\\harddisk11\\partition14
Fibre - 1	ORA_CUSTOMER2_9	\\device\\harddisk11\\partition15
Fibre - 1	ORA_CUSTOMER3_9	\\device\\harddisk11\\partition16
Fibre - 1	ORA_CUSTOMER4_9	\\device\\harddisk11\\partition17
Fibre - 1	ORA_CUSTOMER5_9	\\device\\harddisk11\\partition18
Fibre - 1	ORA_TEMP_10	\\device\\harddisk12\\partition2
Fibre - 1	ORA_IND5_10	\\device\\harddisk12\\partition1
Fibre - 1	ORA_IND1_10	\\device\\harddisk12\\partition3
Fibre - 1	ORA_LINEITEM_10	\\device\\harddisk12\\partition4
Fibre - 1	ORA_LINEITEM_50	\\device\\harddisk12\\partition5
Fibre - 1	ORA_IND3_10	\\device\\harddisk12\\partition6
Fibre - 1	ORA_PS_10	\\device\\harddisk12\\partition7
Fibre - 1	ORA_IND4_10	\\device\\harddisk12\\partition8
Fibre - 1	ORA_ORDER_10	\\device\\harddisk12\\partition9
Fibre - 1	ORA_ORDER_50	\\device\\harddisk12\\partition10

Fibre - 1	ORA_PARTS_10	\device\harddisk12\partition11
Fibre - 1	ORA_PARTS_50	\device\harddisk12\partition12
Fibre - 1	ORA_SUPPLIER_10	\device\harddisk12\partition13
Fibre - 1	ORA_CUSTOMER1_10	\device\harddisk12\partition14
Fibre - 1	ORA_CUSTOMER2_10	\device\harddisk12\partition15
Fibre - 1	ORA_CUSTOMER3_10	\device\harddisk12\partition16
Fibre - 1	ORA_CUSTOMER4_10	\device\harddisk12\partition17
Fibre - 1	ORA_CUSTOMER5_10	\device\harddisk12\partition18
Fibre - 2	ORA_TEMP_11	\device\harddisk13\partition2
Fibre - 2	ORA_IND5_11	\device\harddisk13\partition1
Fibre - 2	ORA_IND1_11	\device\harddisk13\partition3
Fibre - 2	ORA_LINEITEM_11	\device\harddisk13\partition4
Fibre - 2	ORA_LINEITEM_51	\device\harddisk13\partition5
Fibre - 2	ORA_IND3_11	\device\harddisk13\partition6
Fibre - 2	ORA_PS_11	\device\harddisk13\partition7
Fibre - 2	ORA_IND4_11	\device\harddisk13\partition8
Fibre - 2	ORA_ORDER_11	\device\harddisk13\partition9
Fibre - 2	ORA_ORDER_51	\device\harddisk13\partition10
Fibre - 2	ORA_PARTS_11	\device\harddisk13\partition11
Fibre - 2	ORA_SUPPLIER_11	\device\harddisk13\partition12
Fibre - 2	ORA_CUSTOMER1_11	\device\harddisk13\partition13
Fibre - 2	ORA_CUSTOMER2_11	\device\harddisk13\partition14
Fibre - 2	ORA_CUSTOMER3_11	\device\harddisk13\partition15
Fibre - 2	ORA_CUSTOMER4_11	\device\harddisk13\partition16
Fibre - 2	ORA_CUSTOMER5_11	\device\harddisk13\partition17
Fibre - 2	ORA_TEMP_12	\device\harddisk14\partition2
Fibre - 2	ORA_IND5_12	\device\harddisk14\partition1
Fibre - 2	ORA_IND1_12	\device\harddisk14\partition3
Fibre - 2	ORA_LINEITEM_12	\device\harddisk14\partition4
Fibre - 2	ORA_LINEITEM_52	\device\harddisk14\partition5
Fibre - 2	ORA_IND3_12	\device\harddisk14\partition6
Fibre - 2	ORA_PS_12	\device\harddisk14\partition7
Fibre - 2	ORA_IND4_12	\device\harddisk14\partition8
Fibre - 2	ORA_ORDER_12	\device\harddisk14\partition9
Fibre - 2	ORA_ORDER_52	\device\harddisk14\partition10
Fibre - 2	ORA_PARTS_12	\device\harddisk14\partition11
Fibre - 2	ORA_SUPPLIER_12	\device\harddisk14\partition12
Fibre - 2	ORA_CUSTOMER1_12	\device\harddisk14\partition13
Fibre - 2	ORA_CUSTOMER2_12	\device\harddisk14\partition14
Fibre - 2	ORA_CUSTOMER3_12	\device\harddisk14\partition15
Fibre - 2	ORA_CUSTOMER4_12	\device\harddisk14\partition16
Fibre - 2	ORA_CUSTOMER5_12	\device\harddisk14\partition17
Fibre - 2	ORA_TEMP_13	\device\harddisk15\partition2
Fibre - 2	ORA_IND5_13	\device\harddisk15\partition1
Fibre - 2	ORA_IND1_13	\device\harddisk15\partition3
Fibre - 2	ORA_LINEITEM_13	\device\harddisk15\partition4
Fibre - 2	ORA_LINEITEM_53	\device\harddisk15\partition5
Fibre - 2	ORA_IND3_13	\device\harddisk15\partition6
Fibre - 2	ORA_PS_13	\device\harddisk15\partition7

Fibre - 2	ORA_IND4_13	\device\harddisk15\partition8
Fibre - 2	ORA_ORDER_13	\device\harddisk15\partition9
Fibre - 2	ORA_ORDER_53	\device\harddisk15\partition10
Fibre - 2	ORA_PARTS_13	\device\harddisk15\partition11
Fibre - 2	ORA_SUPPLIER_13	\device\harddisk15\partition12
Fibre - 2	ORA_CUSTOMER1_13	\device\harddisk15\partition13
Fibre - 2	ORA_CUSTOMER2_13	\device\harddisk15\partition14
Fibre - 2	ORA_CUSTOMER3_13	\device\harddisk15\partition15
Fibre - 2	ORA_CUSTOMER4_13	\device\harddisk15\partition16
Fibre - 2	ORA_CUSTOMER5_13	\device\harddisk15\partition17
Fibre - 2	ORA_TEMP_14	\device\harddisk16\partition2
Fibre - 2	ORA_IND5_14	\device\harddisk16\partition1
Fibre - 2	ORA_IND1_14	\device\harddisk16\partition3
Fibre - 2	ORA_LINEITEM_14	\device\harddisk16\partition4
Fibre - 2	ORA_LINEITEM_54	\device\harddisk16\partition5
Fibre - 2	ORA_IND3_14	\device\harddisk16\partition6
Fibre - 2	ORA_PS_14	\device\harddisk16\partition7
Fibre - 2	ORA_IND4_14	\device\harddisk16\partition8
Fibre - 2	ORA_ORDER_14	\device\harddisk16\partition9
Fibre - 2	ORA_ORDER_54	\device\harddisk16\partition10
Fibre - 2	ORA_PARTS_14	\device\harddisk16\partition11
Fibre - 2	ORA_SUPPLIER_14	\device\harddisk16\partition12
Fibre - 2	ORA_CUSTOMER1_14	\device\harddisk16\partition13
Fibre - 2	ORA_CUSTOMER2_14	\device\harddisk16\partition14
Fibre - 2	ORA_CUSTOMER3_14	\device\harddisk16\partition15
Fibre - 2	ORA_CUSTOMER4_14	\device\harddisk16\partition16
Fibre - 2	ORA_CUSTOMER5_14	\device\harddisk16\partition17
Fibre - 2	ORA_TEMP_15	\device\harddisk17\partition2
Fibre - 2	ORA_IND5_15	\device\harddisk17\partition1
Fibre - 2	ORA_IND1_15	\device\harddisk17\partition3
Fibre - 2	ORA_LINEITEM_15	\device\harddisk17\partition4
Fibre - 2	ORA_LINEITEM_55	\device\harddisk17\partition5
Fibre - 2	ORA_IND3_15	\device\harddisk17\partition6
Fibre - 2	ORA_PS_15	\device\harddisk17\partition7
Fibre - 2	ORA_IND4_15	\device\harddisk17\partition8
Fibre - 2	ORA_ORDER_15	\device\harddisk17\partition9
Fibre - 2	ORA_ORDER_55	\device\harddisk17\partition10
Fibre - 2	ORA_PARTS_15	\device\harddisk17\partition11
Fibre - 2	ORA_SUPPLIER_15	\device\harddisk17\partition12
Fibre - 2	ORA_CUSTOMER1_15	\device\harddisk17\partition13
Fibre - 2	ORA_CUSTOMER2_15	\device\harddisk17\partition14
Fibre - 2	ORA_CUSTOMER3_15	\device\harddisk17\partition15
Fibre - 2	ORA_CUSTOMER4_15	\device\harddisk17\partition16
Fibre - 2	ORA_CUSTOMER5_15	\device\harddisk17\partition17
Fibre - 2	ORA_TEMP_16	\device\harddisk18\partition2
Fibre - 2	ORA_IND5_16	\device\harddisk18\partition1
Fibre - 2	ORA_IND1_16	\device\harddisk18\partition3
Fibre - 2	ORA_LINEITEM_16	\device\harddisk18\partition4
Fibre - 2	ORA_LINEITEM_56	\device\harddisk18\partition5

Fibre - 2	ORA_IND3_16	\\device\\harddisk18\\partition6
Fibre - 2	ORA_PS_16	\\device\\harddisk18\\partition7
Fibre - 2	ORA_IND4_16	\\device\\harddisk18\\partition8
Fibre - 2	ORA_ORDER_16	\\device\\harddisk18\\partition9
Fibre - 2	ORA_ORDER_56	\\device\\harddisk18\\partition10
Fibre - 2	ORA_PARTS_16	\\device\\harddisk18\\partition11
Fibre - 2	ORA_SUPPLIER_16	\\device\\harddisk18\\partition12
Fibre - 2	ORA_CUSTOMER1_16	\\device\\harddisk18\\partition13
Fibre - 2	ORA_CUSTOMER2_16	\\device\\harddisk18\\partition14
Fibre - 2	ORA_CUSTOMER3_16	\\device\\harddisk18\\partition15
Fibre - 2	ORA_CUSTOMER4_16	\\device\\harddisk18\\partition16
Fibre - 2	ORA_CUSTOMER5_16	\\device\\harddisk18\\partition17
Fibre - 3	ORA_TEMP_17	\\device\\harddisk19\\partition2
Fibre - 3	ORA_IND5_17	\\device\\harddisk19\\partition1
Fibre - 3	ORA_IND1_17	\\device\\harddisk19\\partition3
Fibre - 3	ORA_LINEITEM_17	\\device\\harddisk19\\partition4
Fibre - 3	ORA_LINEITEM_57	\\device\\harddisk19\\partition5
Fibre - 3	ORA_IND3_17	\\device\\harddisk19\\partition6
Fibre - 3	ORA_PS_17	\\device\\harddisk19\\partition7
Fibre - 3	ORA_IND4_17	\\device\\harddisk19\\partition8
Fibre - 3	ORA_ORDER_17	\\device\\harddisk19\\partition9
Fibre - 3	ORA_ORDER_57	\\device\\harddisk19\\partition10
Fibre - 3	ORA_PARTS_17	\\device\\harddisk19\\partition11
Fibre - 3	ORA_SUPPLIER_17	\\device\\harddisk19\\partition12
Fibre - 3	ORA_CUSTOMER1_17	\\device\\harddisk19\\partition13
Fibre - 3	ORA_CUSTOMER2_17	\\device\\harddisk19\\partition14
Fibre - 3	ORA_CUSTOMER3_17	\\device\\harddisk19\\partition15
Fibre - 3	ORA_CUSTOMER4_17	\\device\\harddisk19\\partition16
Fibre - 3	ORA_CUSTOMER5_17	\\device\\harddisk19\\partition17
Fibre - 3	ORA_TEMP_18	\\device\\harddisk20\\partition2
Fibre - 3	ORA_IND5_18	\\device\\harddisk20\\partition1
Fibre - 3	ORA_IND1_18	\\device\\harddisk20\\partition3
Fibre - 3	ORA_LINEITEM_18	\\device\\harddisk20\\partition4
Fibre - 3	ORA_LINEITEM_58	\\device\\harddisk20\\partition5
Fibre - 3	ORA_IND3_18	\\device\\harddisk20\\partition6
Fibre - 3	ORA_PS_18	\\device\\harddisk20\\partition7
Fibre - 3	ORA_IND4_18	\\device\\harddisk20\\partition8
Fibre - 3	ORA_ORDER_18	\\device\\harddisk20\\partition9
Fibre - 3	ORA_ORDER_58	\\device\\harddisk20\\partition10
Fibre - 3	ORA_PARTS_18	\\device\\harddisk20\\partition11
Fibre - 3	ORA_SUPPLIER_18	\\device\\harddisk20\\partition12
Fibre - 3	ORA_CUSTOMER1_18	\\device\\harddisk20\\partition13
Fibre - 3	ORA_CUSTOMER2_18	\\device\\harddisk20\\partition14
Fibre - 3	ORA_CUSTOMER3_18	\\device\\harddisk20\\partition15
Fibre - 3	ORA_CUSTOMER4_18	\\device\\harddisk20\\partition16
Fibre - 3	ORA_CUSTOMER5_18	\\device\\harddisk20\\partition17
Fibre - 3	ORA_TEMP_19	\\device\\harddisk21\\partition2
Fibre - 3	ORA_IND5_19	\\device\\harddisk21\\partition1
Fibre - 3	ORA_IND1_19	\\device\\harddisk21\\partition3

Fibre - 3	ORA_LINEITEM_19	\device\harddisk21\partition4
Fibre - 3	ORA_LINEITEM_59	\device\harddisk21\partition5
Fibre - 3	ORA_IND3_19	\device\harddisk21\partition6
Fibre - 3	ORA_PS_19	\device\harddisk21\partition7
Fibre - 3	ORA_IND4_19	\device\harddisk21\partition8
Fibre - 3	ORA_ORDER_19	\device\harddisk21\partition9
Fibre - 3	ORA_ORDER_59	\device\harddisk21\partition10
Fibre - 3	ORA_PARTS_19	\device\harddisk21\partition11
Fibre - 3	ORA_SUPPLIER_19	\device\harddisk21\partition12
Fibre - 3	ORA_CUSTOMER1_19	\device\harddisk21\partition13
Fibre - 3	ORA_CUSTOMER2_19	\device\harddisk21\partition14
Fibre - 3	ORA_CUSTOMER3_19	\device\harddisk21\partition15
Fibre - 3	ORA_CUSTOMER4_19	\device\harddisk21\partition16
Fibre - 3	ORA_CUSTOMER5_19	\device\harddisk21\partition17
Fibre - 3	ORA_TEMP_20	\device\harddisk22\partition2
Fibre - 3	ORA_IND5_20	\device\harddisk22\partition1
Fibre - 3	ORA_IND1_20	\device\harddisk22\partition3
Fibre - 3	ORA_LINEITEM_20	\device\harddisk22\partition4
Fibre - 3	ORA_LINEITEM_60	\device\harddisk22\partition5
Fibre - 3	ORA_IND3_20	\device\harddisk22\partition6
Fibre - 3	ORA_PS_20	\device\harddisk22\partition7
Fibre - 3	ORA_IND4_20	\device\harddisk22\partition8
Fibre - 3	ORA_ORDER_20	\device\harddisk22\partition9
Fibre - 3	ORA_ORDER_60	\device\harddisk22\partition10
Fibre - 3	ORA_PARTS_20	\device\harddisk22\partition11
Fibre - 3	ORA_SUPPLIER_20	\device\harddisk22\partition12
Fibre - 3	ORA_CUSTOMER1_20	\device\harddisk22\partition13
Fibre - 3	ORA_CUSTOMER2_20	\device\harddisk22\partition14
Fibre - 3	ORA_CUSTOMER3_20	\device\harddisk22\partition15
Fibre - 3	ORA_CUSTOMER4_20	\device\harddisk22\partition16
Fibre - 3	ORA_CUSTOMER5_20	\device\harddisk22\partition17
Fibre - 3	ORA_TEMP_21	\device\harddisk23\partition2
Fibre - 3	ORA_IND5_21	\device\harddisk23\partition1
Fibre - 3	ORA_IND1_21	\device\harddisk23\partition3
Fibre - 3	ORA_LINEITEM_21	\device\harddisk23\partition4
Fibre - 3	ORA_LINEITEM_61	\device\harddisk23\partition5
Fibre - 3	ORA_IND3_21	\device\harddisk23\partition6
Fibre - 3	ORA_PS_21	\device\harddisk23\partition7
Fibre - 3	ORA_IND4_21	\device\harddisk23\partition8
Fibre - 3	ORA_ORDER_21	\device\harddisk23\partition9
Fibre - 3	ORA_ORDER_61	\device\harddisk23\partition10
Fibre - 3	ORA_PARTS_21	\device\harddisk23\partition11
Fibre - 3	ORA_SUPPLIER_21	\device\harddisk23\partition12
Fibre - 3	ORA_CUSTOMER1_21	\device\harddisk23\partition13
Fibre - 3	ORA_CUSTOMER2_21	\device\harddisk23\partition14
Fibre - 3	ORA_CUSTOMER3_21	\device\harddisk23\partition15
Fibre - 3	ORA_CUSTOMER4_21	\device\harddisk23\partition16
Fibre - 3	ORA_CUSTOMER5_21	\device\harddisk23\partition17
Fibre - 3	ORA_TEMP_22	\device\harddisk24\partition2

Fibre - 3	ORA_IND5_22	\\device\\harddisk24\\partition1
Fibre - 3	ORA_IND1_22	\\device\\harddisk24\\partition3
Fibre - 3	ORA_LINEITEM_22	\\device\\harddisk24\\partition4
Fibre - 3	ORA_LINEITEM_62	\\device\\harddisk24\\partition5
Fibre - 3	ORA_IND3_22	\\device\\harddisk24\\partition6
Fibre - 3	ORA_PS_22	\\device\\harddisk24\\partition7
Fibre - 3	ORA_IND4_22	\\device\\harddisk24\\partition8
Fibre - 3	ORA_ORDER_22	\\device\\harddisk24\\partition9
Fibre - 3	ORA_ORDER_62	\\device\\harddisk24\\partition10
Fibre - 3	ORA_PARTS_22	\\device\\harddisk24\\partition11
Fibre - 3	ORA_SUPPLIER_22	\\device\\harddisk24\\partition12
Fibre - 3	ORA_CUSTOMER1_22	\\device\\harddisk24\\partition13
Fibre - 3	ORA_CUSTOMER2_22	\\device\\harddisk24\\partition14
Fibre - 3	ORA_CUSTOMER3_22	\\device\\harddisk24\\partition15
Fibre - 3	ORA_CUSTOMER4_22	\\device\\harddisk24\\partition16
Fibre - 3	ORA_CUSTOMER5_22	\\device\\harddisk24\\partition17
Fibre - 4	ORA_TEMP_23	\\device\\harddisk25\\partition2
Fibre - 4	ORA_IND5_23	\\device\\harddisk25\\partition1
Fibre - 4	ORA_IND1_23	\\device\\harddisk25\\partition3
Fibre - 4	ORA_LINEITEM_23	\\device\\harddisk25\\partition4
Fibre - 4	ORA_LINEITEM_63	\\device\\harddisk25\\partition5
Fibre - 4	ORA_IND3_23	\\device\\harddisk25\\partition6
Fibre - 4	ORA_PS_23	\\device\\harddisk25\\partition7
Fibre - 4	ORA_IND4_23	\\device\\harddisk25\\partition8
Fibre - 4	ORA_ORDER_23	\\device\\harddisk25\\partition9
Fibre - 4	ORA_ORDER_63	\\device\\harddisk25\\partition10
Fibre - 4	ORA_PARTS_23	\\device\\harddisk25\\partition11
Fibre - 4	ORA_SUPPLIER_23	\\device\\harddisk25\\partition12
Fibre - 4	ORA_CUSTOMER1_23	\\device\\harddisk25\\partition13
Fibre - 4	ORA_CUSTOMER2_23	\\device\\harddisk25\\partition14
Fibre - 4	ORA_CUSTOMER3_23	\\device\\harddisk25\\partition15
Fibre - 4	ORA_CUSTOMER4_23	\\device\\harddisk25\\partition16
Fibre - 4	ORA_CUSTOMER5_23	\\device\\harddisk25\\partition17
Fibre - 4	ORA_TEMP_24	\\device\\harddisk26\\partition2
Fibre - 4	ORA_IND5_24	\\device\\harddisk26\\partition1
Fibre - 4	ORA_IND1_24	\\device\\harddisk26\\partition3
Fibre - 4	ORA_LINEITEM_24	\\device\\harddisk26\\partition4
Fibre - 4	ORA_LINEITEM_64	\\device\\harddisk26\\partition5
Fibre - 4	ORA_IND3_24	\\device\\harddisk26\\partition6
Fibre - 4	ORA_PS_24	\\device\\harddisk26\\partition7
Fibre - 4	ORA_IND4_24	\\device\\harddisk26\\partition8
Fibre - 4	ORA_ORDER_24	\\device\\harddisk26\\partition9
Fibre - 4	ORA_ORDER_64	\\device\\harddisk26\\partition10
Fibre - 4	ORA_PARTS_24	\\device\\harddisk26\\partition11
Fibre - 4	ORA_SUPPLIER_24	\\device\\harddisk26\\partition12
Fibre - 4	ORA_CUSTOMER1_24	\\device\\harddisk26\\partition13
Fibre - 4	ORA_CUSTOMER2_24	\\device\\harddisk26\\partition14
Fibre - 4	ORA_CUSTOMER3_24	\\device\\harddisk26\\partition15
Fibre - 4	ORA_CUSTOMER4_24	\\device\\harddisk26\\partition16

Fibre - 4	ORA_CUSTOMER5_24	\device\harddisk26\partition17
Fibre - 4	ORA_TEMP_25	\device\harddisk27\partition2
Fibre - 4	ORA_IND5_25	\device\harddisk27\partition1
Fibre - 4	ORA_IND1_25	\device\harddisk27\partition3
Fibre - 4	ORA_LINEITEM_25	\device\harddisk27\partition4
Fibre - 4	ORA_LINEITEM_65	\device\harddisk27\partition5
Fibre - 4	ORA_IND3_25	\device\harddisk27\partition6
Fibre - 4	ORA_PS_25	\device\harddisk27\partition7
Fibre - 4	ORA_IND4_25	\device\harddisk27\partition8
Fibre - 4	ORA_ORDER_25	\device\harddisk27\partition9
Fibre - 4	ORA_ORDER_65	\device\harddisk27\partition10
Fibre - 4	ORA_PARTS_25	\device\harddisk27\partition11
Fibre - 4	ORA_SUPPLIER_25	\device\harddisk27\partition12
Fibre - 4	ORA_CUSTOMER1_25	\device\harddisk27\partition13
Fibre - 4	ORA_CUSTOMER2_25	\device\harddisk27\partition14
Fibre - 4	ORA_CUSTOMER3_25	\device\harddisk27\partition15
Fibre - 4	ORA_CUSTOMER4_25	\device\harddisk27\partition16
Fibre - 4	ORA_CUSTOMER5_25	\device\harddisk27\partition17
Fibre - 4	ORA_TEMP_26	\device\harddisk28\partition2
Fibre - 4	ORA_IND5_26	\device\harddisk28\partition1
Fibre - 4	ORA_IND1_26	\device\harddisk28\partition3
Fibre - 4	ORA_LINEITEM_26	\device\harddisk28\partition4
Fibre - 4	ORA_LINEITEM_66	\device\harddisk28\partition5
Fibre - 4	ORA_IND3_26	\device\harddisk28\partition6
Fibre - 4	ORA_PS_26	\device\harddisk28\partition7
Fibre - 4	ORA_IND4_26	\device\harddisk28\partition8
Fibre - 4	ORA_ORDER_26	\device\harddisk28\partition9
Fibre - 4	ORA_ORDER_66	\device\harddisk28\partition10
Fibre - 4	ORA_PARTS_26	\device\harddisk28\partition11
Fibre - 4	ORA_SUPPLIER_26	\device\harddisk28\partition12
Fibre - 4	ORA_CUSTOMER1_26	\device\harddisk28\partition13
Fibre - 4	ORA_CUSTOMER2_26	\device\harddisk28\partition14
Fibre - 4	ORA_CUSTOMER3_26	\device\harddisk28\partition15
Fibre - 4	ORA_CUSTOMER4_26	\device\harddisk28\partition16
Fibre - 4	ORA_CUSTOMER5_26	\device\harddisk28\partition17
Fibre - 4	ORA_TEMP_27	\device\harddisk29\partition2
Fibre - 4	ORA_IND5_27	\device\harddisk29\partition1
Fibre - 4	ORA_IND1_27	\device\harddisk29\partition3
Fibre - 4	ORA_LINEITEM_27	\device\harddisk29\partition4
Fibre - 4	ORA_LINEITEM_67	\device\harddisk29\partition5
Fibre - 4	ORA_IND3_27	\device\harddisk29\partition6
Fibre - 4	ORA_PS_27	\device\harddisk29\partition7
Fibre - 4	ORA_IND4_27	\device\harddisk29\partition8
Fibre - 4	ORA_ORDER_27	\device\harddisk29\partition9
Fibre - 4	ORA_ORDER_67	\device\harddisk29\partition10
Fibre - 4	ORA_PARTS_27	\device\harddisk29\partition11
Fibre - 4	ORA_SUPPLIER_27	\device\harddisk29\partition12
Fibre - 4	ORA_CUSTOMER1_27	\device\harddisk29\partition13
Fibre - 4	ORA_CUSTOMER2_27	\device\harddisk29\partition14

Fibre - 4	ORA_CUSTOMER3_27	\\device\\harddisk29\\partition15
Fibre - 4	ORA_CUSTOMER4_27	\\device\\harddisk29\\partition16
Fibre - 4	ORA_CUSTOMER5_27	\\device\\harddisk29\\partition17
Fibre - 4	ORA_TEMP_28	\\device\\harddisk30\\partition2
Fibre - 4	ORA_IND5_28	\\device\\harddisk30\\partition1
Fibre - 4	ORA_IND1_28	\\device\\harddisk30\\partition3
Fibre - 4	ORA_LINEITEM_28	\\device\\harddisk30\\partition4
Fibre - 4	ORA_LINEITEM_68	\\device\\harddisk30\\partition5
Fibre - 4	ORA_IND3_28	\\device\\harddisk30\\partition6
Fibre - 4	ORA_PS_28	\\device\\harddisk30\\partition7
Fibre - 4	ORA_IND4_28	\\device\\harddisk30\\partition8
Fibre - 4	ORA_ORDER_28	\\device\\harddisk30\\partition9
Fibre - 4	ORA_ORDER_68	\\device\\harddisk30\\partition10
Fibre - 4	ORA_PARTS_28	\\device\\harddisk30\\partition11
Fibre - 4	ORA_SUPPLIER_28	\\device\\harddisk30\\partition12
Fibre - 4	ORA_CUSTOMER1_28	\\device\\harddisk30\\partition13
Fibre - 4	ORA_CUSTOMER2_28	\\device\\harddisk30\\partition14
Fibre - 4	ORA_CUSTOMER3_28	\\device\\harddisk30\\partition15
Fibre - 4	ORA_CUSTOMER4_28	\\device\\harddisk30\\partition16
Fibre - 4	ORA_CUSTOMER5_28	\\device\\harddisk30\\partition17
Fibre - 5	ORA_TEMP_29	\\device\\harddisk31\\partition2
Fibre - 5	ORA_IND5_29	\\device\\harddisk31\\partition1
Fibre - 5	ORA_IND1_29	\\device\\harddisk31\\partition3
Fibre - 5	ORA_LINEITEM_29	\\device\\harddisk31\\partition4
Fibre - 5	ORA_LINEITEM_69	\\device\\harddisk31\\partition5
Fibre - 5	ORA_IND3_29	\\device\\harddisk31\\partition6
Fibre - 5	ORA_PS_29	\\device\\harddisk31\\partition7
Fibre - 5	ORA_IND4_29	\\device\\harddisk31\\partition8
Fibre - 5	ORA_ORDER_29	\\device\\harddisk31\\partition9
Fibre - 5	ORA_ORDER_69	\\device\\harddisk31\\partition10
Fibre - 5	ORA_PARTS_29	\\device\\harddisk31\\partition11
Fibre - 5	ORA_SUPPLIER_29	\\device\\harddisk31\\partition12
Fibre - 5	ORA_CUSTOMER1_29	\\device\\harddisk31\\partition13
Fibre - 5	ORA_CUSTOMER2_29	\\device\\harddisk31\\partition14
Fibre - 5	ORA_CUSTOMER3_29	\\device\\harddisk31\\partition15
Fibre - 5	ORA_CUSTOMER4_29	\\device\\harddisk31\\partition16
Fibre - 5	ORA_CUSTOMER5_29	\\device\\harddisk31\\partition17
Fibre - 5	ORA_TEMP_30	\\device\\harddisk32\\partition2
Fibre - 5	ORA_IND5_30	\\device\\harddisk32\\partition1
Fibre - 5	ORA_IND1_30	\\device\\harddisk32\\partition3
Fibre - 5	ORA_LINEITEM_30	\\device\\harddisk32\\partition4
Fibre - 5	ORA_LINEITEM_70	\\device\\harddisk32\\partition5
Fibre - 5	ORA_IND3_30	\\device\\harddisk32\\partition6
Fibre - 5	ORA_PS_30	\\device\\harddisk32\\partition7
Fibre - 5	ORA_IND4_30	\\device\\harddisk32\\partition8
Fibre - 5	ORA_ORDER_30	\\device\\harddisk32\\partition9
Fibre - 5	ORA_ORDER_70	\\device\\harddisk32\\partition10
Fibre - 5	ORA_PARTS_30	\\device\\harddisk32\\partition11
Fibre - 5	ORA_SUPPLIER_30	\\device\\harddisk32\\partition12

Fibre - 5	ORA_CUSTOMER1_30	\device\harddisk32\partition13
Fibre - 5	ORA_CUSTOMER2_30	\device\harddisk32\partition14
Fibre - 5	ORA_CUSTOMER3_30	\device\harddisk32\partition15
Fibre - 5	ORA_CUSTOMER4_30	\device\harddisk32\partition16
Fibre - 5	ORA_CUSTOMER5_30	\device\harddisk32\partition17
Fibre - 5	ORA_TEMP_31	\device\harddisk33\partition2
Fibre - 5	ORA_IND5_31	\device\harddisk33\partition1
Fibre - 5	ORA_IND1_31	\device\harddisk33\partition3
Fibre - 5	ORA_LINEITEM_31	\device\harddisk33\partition4
Fibre - 5	ORA_LINEITEM_71	\device\harddisk33\partition5
Fibre - 5	ORA_IND3_31	\device\harddisk33\partition6
Fibre - 5	ORA_PS_31	\device\harddisk33\partition7
Fibre - 5	ORA_IND4_31	\device\harddisk33\partition8
Fibre - 5	ORA_ORDER_31	\device\harddisk33\partition9
Fibre - 5	ORA_ORDER_71	\device\harddisk33\partition10
Fibre - 5	ORA_PARTS_31	\device\harddisk33\partition11
Fibre - 5	ORA_SUPPLIER_31	\device\harddisk33\partition12
Fibre - 5	ORA_CUSTOMER1_31	\device\harddisk33\partition13
Fibre - 5	ORA_CUSTOMER2_31	\device\harddisk33\partition14
Fibre - 5	ORA_CUSTOMER3_31	\device\harddisk33\partition15
Fibre - 5	ORA_CUSTOMER4_31	\device\harddisk33\partition16
Fibre - 5	ORA_CUSTOMER5_31	\device\harddisk33\partition17
Fibre - 5	ORA_TEMP_32	\device\harddisk34\partition2
Fibre - 5	ORA_IND5_32	\device\harddisk34\partition1
Fibre - 5	ORA_IND1_32	\device\harddisk34\partition3
Fibre - 5	ORA_LINEITEM_32	\device\harddisk34\partition4
Fibre - 5	ORA_LINEITEM_72	\device\harddisk34\partition5
Fibre - 5	ORA_IND3_32	\device\harddisk34\partition6
Fibre - 5	ORA_PS_32	\device\harddisk34\partition7
Fibre - 5	ORA_IND4_32	\device\harddisk34\partition8
Fibre - 5	ORA_ORDER_32	\device\harddisk34\partition9
Fibre - 5	ORA_ORDER_72	\device\harddisk34\partition10
Fibre - 5	ORA_PARTS_32	\device\harddisk34\partition11
Fibre - 5	ORA_SUPPLIER_32	\device\harddisk34\partition12
Fibre - 5	ORA_CUSTOMER1_32	\device\harddisk34\partition13
Fibre - 5	ORA_CUSTOMER2_32	\device\harddisk34\partition14
Fibre - 5	ORA_CUSTOMER3_32	\device\harddisk34\partition15
Fibre - 5	ORA_CUSTOMER4_32	\device\harddisk34\partition16
Fibre - 5	ORA_CUSTOMER5_32	\device\harddisk34\partition17
Fibre - 5	ORA_TEMP_33	\device\harddisk35\partition2
Fibre - 5	ORA_IND5_33	\device\harddisk35\partition1
Fibre - 5	ORA_IND1_33	\device\harddisk35\partition3
Fibre - 5	ORA_LINEITEM_33	\device\harddisk35\partition4
Fibre - 5	ORA_LINEITEM_73	\device\harddisk35\partition5
Fibre - 5	ORA_IND3_33	\device\harddisk35\partition6
Fibre - 5	ORA_PS_33	\device\harddisk35\partition7
Fibre - 5	ORA_IND4_33	\device\harddisk35\partition8
Fibre - 5	ORA_ORDER_33	\device\harddisk35\partition9
Fibre - 5	ORA_ORDER_73	\device\harddisk35\partition10

Fibre - 5	ORA_PARTS_33	\\device\\harddisk35\\partition11
Fibre - 5	ORA_SUPPLIER_33	\\device\\harddisk35\\partition12
Fibre - 5	ORA_CUSTOMER1_33	\\device\\harddisk35\\partition13
Fibre - 5	ORA_CUSTOMER2_33	\\device\\harddisk35\\partition14
Fibre - 5	ORA_CUSTOMER3_33	\\device\\harddisk35\\partition15
Fibre - 5	ORA_CUSTOMER4_33	\\device\\harddisk35\\partition16
Fibre - 5	ORA_CUSTOMER5_33	\\device\\harddisk35\\partition17
Fibre - 5	ORA_TEMP_34	\\device\\harddisk36\\partition2
Fibre - 5	ORA_IND5_34	\\device\\harddisk36\\partition1
Fibre - 5	ORA_IND1_34	\\device\\harddisk36\\partition3
Fibre - 5	ORA_LINEITEM_34	\\device\\harddisk36\\partition4
Fibre - 5	ORA_LINEITEM_74	\\device\\harddisk36\\partition5
Fibre - 5	ORA_IND3_34	\\device\\harddisk36\\partition6
Fibre - 5	ORA_PS_34	\\device\\harddisk36\\partition7
Fibre - 5	ORA_IND4_34	\\device\\harddisk36\\partition8
Fibre - 5	ORA_ORDER_34	\\device\\harddisk36\\partition9
Fibre - 5	ORA_ORDER_74	\\device\\harddisk36\\partition10
Fibre - 5	ORA_PARTS_34	\\device\\harddisk36\\partition11
Fibre - 5	ORA_SUPPLIER_34	\\device\\harddisk36\\partition12
Fibre - 5	ORA_CUSTOMER1_34	\\device\\harddisk36\\partition13
Fibre - 5	ORA_CUSTOMER2_34	\\device\\harddisk36\\partition14
Fibre - 5	ORA_CUSTOMER3_34	\\device\\harddisk36\\partition15
Fibre - 5	ORA_CUSTOMER4_34	\\device\\harddisk36\\partition16
Fibre - 5	ORA_CUSTOMER5_34	\\device\\harddisk36\\partition17
Fibre - 6	ORA_TEMP_35	\\device\\harddisk37\\partition2
Fibre - 6	ORA_IND5_35	\\device\\harddisk37\\partition1
Fibre - 6	ORA_IND1_35	\\device\\harddisk37\\partition3
Fibre - 6	ORA_LINEITEM_35	\\device\\harddisk37\\partition4
Fibre - 6	ORA_LINEITEM_75	\\device\\harddisk37\\partition5
Fibre - 6	ORA_IND3_35	\\device\\harddisk37\\partition6
Fibre - 6	ORA_PS_35	\\device\\harddisk37\\partition7
Fibre - 6	ORA_IND4_35	\\device\\harddisk37\\partition8
Fibre - 6	ORA_ORDER_35	\\device\\harddisk37\\partition9
Fibre - 6	ORA_ORDER_75	\\device\\harddisk37\\partition10
Fibre - 6	ORA_PARTS_35	\\device\\harddisk37\\partition11
Fibre - 6	ORA_SUPPLIER_35	\\device\\harddisk37\\partition12
Fibre - 6	ORA_CUSTOMER1_35	\\device\\harddisk37\\partition13
Fibre - 6	ORA_CUSTOMER2_35	\\device\\harddisk37\\partition14
Fibre - 6	ORA_CUSTOMER3_35	\\device\\harddisk37\\partition15
Fibre - 6	ORA_CUSTOMER4_35	\\device\\harddisk37\\partition16
Fibre - 6	ORA_CUSTOMER5_35	\\device\\harddisk37\\partition17
Fibre - 6	ORA_TEMP_36	\\device\\harddisk38\\partition2
Fibre - 6	ORA_IND5_36	\\device\\harddisk38\\partition1
Fibre - 6	ORA_IND1_36	\\device\\harddisk38\\partition3
Fibre - 6	ORA_LINEITEM_36	\\device\\harddisk38\\partition4
Fibre - 6	ORA_LINEITEM_76	\\device\\harddisk38\\partition5
Fibre - 6	ORA_IND3_36	\\device\\harddisk38\\partition6
Fibre - 6	ORA_PS_36	\\device\\harddisk38\\partition7
Fibre - 6	ORA_IND4_36	\\device\\harddisk38\\partition8

Fibre - 6	ORA_ORDER_36	\device\harddisk38\partition9
Fibre - 6	ORA_ORDER_76	\device\harddisk38\partition10
Fibre - 6	ORA_PARTS_36	\device\harddisk38\partition11
Fibre - 6	ORA_SUPPLIER_36	\device\harddisk38\partition12
Fibre - 6	ORA_CUSTOMER1_36	\device\harddisk38\partition13
Fibre - 6	ORA_CUSTOMER2_36	\device\harddisk38\partition14
Fibre - 6	ORA_CUSTOMER3_36	\device\harddisk38\partition15
Fibre - 6	ORA_CUSTOMER4_36	\device\harddisk38\partition16
Fibre - 6	ORA_CUSTOMER5_36	\device\harddisk38\partition17
Fibre - 6	ORA_TEMP_37	\device\harddisk39\partition2
Fibre - 6	ORA_IND5_37	\device\harddisk39\partition1
Fibre - 6	ORA_IND1_37	\device\harddisk39\partition3
Fibre - 6	ORA_LINEITEM_37	\device\harddisk39\partition4
Fibre - 6	ORA_LINEITEM_77	\device\harddisk39\partition5
Fibre - 6	ORA_IND3_37	\device\harddisk39\partition6
Fibre - 6	ORA_PS_37	\device\harddisk39\partition7
Fibre - 6	ORA_IND4_37	\device\harddisk39\partition8
Fibre - 6	ORA_ORDER_37	\device\harddisk39\partition9
Fibre - 6	ORA_ORDER_77	\device\harddisk39\partition10
Fibre - 6	ORA_PARTS_37	\device\harddisk39\partition11
Fibre - 6	ORA_SUPPLIER_37	\device\harddisk39\partition12
Fibre - 6	ORA_CUSTOMER1_37	\device\harddisk39\partition13
Fibre - 6	ORA_CUSTOMER2_37	\device\harddisk39\partition14
Fibre - 6	ORA_CUSTOMER3_37	\device\harddisk39\partition15
Fibre - 6	ORA_CUSTOMER4_37	\device\harddisk39\partition16
Fibre - 6	ORA_CUSTOMER5_37	\device\harddisk39\partition17
Fibre - 6	ORA_TEMP_38	\device\harddisk40\partition2
Fibre - 6	ORA_IND5_38	\device\harddisk40\partition1
Fibre - 6	ORA_IND1_38	\device\harddisk40\partition3
Fibre - 6	ORA_LINEITEM_38	\device\harddisk40\partition4
Fibre - 6	ORA_LINEITEM_78	\device\harddisk40\partition5
Fibre - 6	ORA_IND3_38	\device\harddisk40\partition6
Fibre - 6	ORA_PS_38	\device\harddisk40\partition7
Fibre - 6	ORA_IND4_38	\device\harddisk40\partition8
Fibre - 6	ORA_ORDER_38	\device\harddisk40\partition9
Fibre - 6	ORA_ORDER_78	\device\harddisk40\partition10
Fibre - 6	ORA_PARTS_38	\device\harddisk40\partition11
Fibre - 6	ORA_SUPPLIER_38	\device\harddisk40\partition12
Fibre - 6	ORA_CUSTOMER1_38	\device\harddisk40\partition13
Fibre - 6	ORA_CUSTOMER2_38	\device\harddisk40\partition14
Fibre - 6	ORA_CUSTOMER3_38	\device\harddisk40\partition15
Fibre - 6	ORA_CUSTOMER4_38	\device\harddisk40\partition16
Fibre - 6	ORA_CUSTOMER5_38	\device\harddisk40\partition17
Fibre - 6	ORA_TEMP_39	\device\harddisk41\partition2
Fibre - 6	ORA_IND5_39	\device\harddisk41\partition1
Fibre - 6	ORA_IND1_39	\device\harddisk41\partition3
Fibre - 6	ORA_LINEITEM_39	\device\harddisk41\partition4
Fibre - 6	ORA_LINEITEM_79	\device\harddisk41\partition5
Fibre - 6	ORA_IND3_39	\device\harddisk41\partition6

Fibre - 6	ORA_PS_39	\\device\\harddisk41\\partition7
Fibre - 6	ORA_IND4_39	\\device\\harddisk41\\partition8
Fibre - 6	ORA_ORDER_39	\\device\\harddisk41\\partition9
Fibre - 6	ORA_ORDER_79	\\device\\harddisk41\\partition10
Fibre - 6	ORA_PARTS_39	\\device\\harddisk41\\partition11
Fibre - 6	ORA_SUPPLIER_39	\\device\\harddisk41\\partition12
Fibre - 6	ORA_CUSTOMER1_39	\\device\\harddisk41\\partition13
Fibre - 6	ORA_CUSTOMER2_39	\\device\\harddisk41\\partition14
Fibre - 6	ORA_CUSTOMER3_39	\\device\\harddisk41\\partition15
Fibre - 6	ORA_CUSTOMER4_39	\\device\\harddisk41\\partition16
Fibre - 6	ORA_CUSTOMER5_39	\\device\\harddisk41\\partition17
Fibre - 6	ORA_TEMP_40	\\device\\harddisk42\\partition2
Fibre - 6	ORA_IND5_40	\\device\\harddisk42\\partition1
Fibre - 6	ORA_IND1_40	\\device\\harddisk42\\partition3
Fibre - 6	ORA_LINEITEM_40	\\device\\harddisk42\\partition4
Fibre - 6	ORA_LINEITEM_80	\\device\\harddisk42\\partition5
Fibre - 6	ORA_IND3_40	\\device\\harddisk42\\partition6
Fibre - 6	ORA_PS_40	\\device\\harddisk42\\partition7
Fibre - 6	ORA_IND4_40	\\device\\harddisk42\\partition8
Fibre - 6	ORA_ORDER_40	\\device\\harddisk42\\partition9
Fibre - 6	ORA_ORDER_80	\\device\\harddisk42\\partition10
Fibre - 6	ORA_PARTS_40	\\device\\harddisk42\\partition11
Fibre - 6	ORA_SUPPLIER_40	\\device\\harddisk42\\partition12
Fibre - 6	ORA_CUSTOMER1_40	\\device\\harddisk42\\partition13
Fibre - 6	ORA_CUSTOMER2_40	\\device\\harddisk42\\partition14
Fibre - 6	ORA_CUSTOMER3_40	\\device\\harddisk42\\partition15
Fibre - 6	ORA_CUSTOMER4_40	\\device\\harddisk42\\partition16
Fibre - 6	ORA_CUSTOMER5_40	\\device\\harddisk42\\partition17

5.3 Partitioning and Replication

The mapping of database partitions/replications must be explicitly described. Implementations may use some form of RAID to ensure high availability. If used for data, auxiliary storage (e.g. indexes) or temporary space, the level of RAID used must be disclosed for each device.

No replication was used for this implementation.

Table 5.3: Distribution of Partitioned Tables

Array Kit	Configuration	Usage	Fraction
Smart 3100 ES	RAID-1	Log1 & Log2	100%
	RAID-1	System, Rollback segments, Nation, Region tables	100%
	RAID-1	Lineitem, orders, parts, partsupp, supplier, customer, Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 2/50 th of parts
	RAID-1	Lineitem, orders, parts, partsupp, supplier, customer, Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 2/50 th of parts
	RAID-1	Lineitem, orders, parts, partsupp, supplier, customer, Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 2/50 th of parts
	RAID-1	Lineitem, orders, parts, partsupp, supplier, customer, Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 2/50 th of parts
1	RAID-1	Lineitem, orders, parts, partsupp, supplier, customer, Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 2/50 th of parts
	RAID-1	Lineitem, orders, parts, partsupp, supplier, customer, Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 2/50 th of parts
	RAID-1	Lineitem, orders, parts, partsupp, supplier, customer and Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 2/50 th of parts
	RAID-1	lineitem, orders, parts, partsupp, supplier, customer and Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 2/50 th of parts
	RAID-1	lineitem, orders, parts, partsupp, supplier, customer and Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 2/50 th of parts
	RAID-1	lineitem, orders, parts, partsupp, supplier, customer and Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 2/50 th of parts
2	RAID-1	lineitem, orders, parts, partsupp, supplier, customer and Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 1/50 th of parts
	RAID-1	lineitem, orders, parts, partsupp, supplier, customer and Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 1/50 th of parts
	RAID-1	lineitem, orders, parts, partsupp,	1/40 th of each of lineitem, orders,

			indexes 1/50 th of parts
	RAID-1	lineitem, orders, parts, partsupp, supplier, customer and Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 1/50 th of parts
	RAID-1	lineitem, orders, parts, partsupp, supplier, customer and Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 1/50 th of parts
	RAID-1	lineitem, orders, parts, partsupp, supplier, customer and Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 1/50 th of parts
	RAID-1	lineitem, orders, parts, partsupp, supplier, customer and Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 1/50 th of parts
6	RAID-1	lineitem, orders, parts, partsupp, supplier, customer and Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 1/50 th of parts
	RAID-1	lineitem, orders, parts, partsupp, supplier, customer and Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 1/50 th of parts
	RAID-1	lineitem, orders, parts, partsupp, supplier, customer and Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 1/50 th of parts
	RAID-1	lineitem, orders, parts, partsupp, supplier, customer and Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 1/50 th of parts
	RAID-1	lineitem, orders, parts, partsupp, supplier, customer and Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 1/50 th of parts
	RAID-1	lineitem, orders, parts, partsupp, supplier, customer and Indexes	1/40 th of each of lineitem, orders, partsupp, supplier, customer, indexes 1/50 th of parts

5.4 DBGEN Version and Modifications

The version number, release number, modification number, and patch level of DBGEN must be disclosed. Any modifications to the DBGEN (see Clause 4.2.1) source code....must be disclosed. In the event that a program other than DBGEN was used to populate the database, it must be disclosed in its entirety..

A few modifications to DBGEN Version 1.2.0 were made for database population. The modified pieces of the DBGEN source code are disclosed in appendix G.

5.5 Database Content of the First Ten Rows

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

Appendix F contains the first ten rows of each table in the test database.

5.6 Database Load time

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

Database load time was 22 hours, 21 minutes and 6 seconds.

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 as defined in 4.1.3.1. The ratio must be reported to the nearest 1/100th, rounded up.

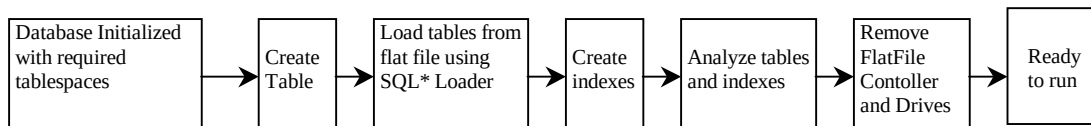
Disk Type	# of Disks	Space per Disk	Sub-total Disk Space	Data Storage Ratio
4.3GB	1	3.99	3.99	
9.1GB	84	8.469	711.396	
		TOTAL	715.386	7.15

5.8 Database Load Mechanism Details and Illustration

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.

DBGEN was used to create flat files which were then loaded into the table spaces using SQL*Loader. The flat files were split into 80 files for the LINEITEM and ORDER tables, 50 files for the PARTS table, 40 files for the rest except the REGION and NATION tables which each had only one file.

KORN shell scripts were used to perform the creation of the tablespaces, tables, and indexes of the database. It was also used to run the analyze scripts.



Clause 5: Performance Metrics and Execution rules Related Items

6.1 Steps in the Power Test

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

The following steps were used to implement the power test:

1. Database Restart
2. UF1 Update Transaction
3. Stream 00 Execution
4. UF2 Update Transaction
5. Reset update functions

6.2 Timing Intervals

The timing intervals (see Clause 5.3.6) for each query of the measured set and for both update functions must be reported for the power test.

The power test timing intervals are presented in the Numerical Quantities Summary in the Preface of this document.

6.3 Number of Streams for The Throughput Test

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

The power test measurements were used for the throughput test. No separate throughput test was run.

6.4 Start/Finish Time of Each Query Stream

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

The throughput test start time and finish time for each stream are reflected in the Numerical Quantities Summary in the Preface of this document.

6.5 Total Elapsed Time

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

The total elapsed time of the throughput test was 11064.85 seconds.

6.6 Start/Finish Time for Update Function

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

The start and finish time for each update function in the update stream are contained in the Numerical Quantities Summary in the Preface of this document.

6.7 Timing Intervals for Each Query and Each Update

The timing intervals (see Clause 5.3.6) for each query of each stream and for each update function must be reported for the throughput test.

The timing intervals for each query and each update function, for the throughput test, are contained in the Numerical Quantities Summary in the Preface of this document. The values for the power test are used for the throughput test.

6.8 Performance Metrics

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

The performance metrics, and the numbers on which they are based with the exception of query 13's timing interval used for the QppD@10 calculation was raised to 3.4 seconds instead of 2.3 to meet the ratio requirements of clause 5.4.1.4, are contained in the Numerical Quantities Summary in the Preface of this document.

6.9 Reproducibility Method

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

Performance results from the first two executions of the TPC-D benchmark indicated the following percent differences for the metrics:

Run	QppD@100GB	QthD@100GB	QphD@100GB
Run 1	1694.6	553.1	968.1
Run 2	1697.0	557.5	972.6
Percent Difference	0.1	0.8	0.5

Clause 6: SUT and Driver Implementation Related Items

7.1 Driver

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

QGEN is first called with a stream ID of 0 to generate the EQT for the power test. The power test is performed by a shell script called **runpower1.sh**. This shell script calls **runuf1.sh** which performs the update function. Queries are then executed with the **qexec.exe** ISL program. The last part of the power test is run through another shell script called **runuf2.sh**. Both wall-clock and high-resolution times are collected for the measurement intervals.

Update functions are parallelized by partitioning the data set and utilizing multiple processes to perform the updates. Two Korn shell programs are written to perform the inserts and deletes required by UF1 and UF2. Logical consistency is ensured by inserting/deleting all of the LINEITEM and ORDERS rows before committing.

7.2 Implementation Specific Layer (ISL)

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

Query execution text generated by QGEN is picked up by the ISL program, **qexec.exe**, which submits the query to the SUT.

The ISL program utilizes the Oracle Call Interface (OCI) to communicate with the Oracle database on the SUT. QETs directly generated by QGEN are read and submitted to the SUT via the ISL program as dynamic SQL statements. The ISL program then fetches the query execution output and reports it to the user. Timings are taken at intervals specified by Clause 5.3.6.2

Clause 7: Pricing Related Items

8.1 Hardware and Software Used in the Priced System

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

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

8.2 Total Five year Price

The total 5-year price of the entire configuration must be reported including: hardware, software, and maintenance charges. Separate component pricing is recommended. The basis of all discounts used must be disclosed.

The total 5-year price of the configuration is \$378,189.

A detailed price sheet of all the hardware and software used in this configuration and the 5-year maintenance cost is included in the executive summary at the beginning of this document.

8.3 Availability Date

The committed delivery date for general availability of products used in the price calculations must be reported. When the priced system includes products with different availability dates, the availability date reported on 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..

All components are available as of the date reported in the executive summary of this report.

Clause 8: Related Items

9.1 Auditor's Report & Attestation Letter

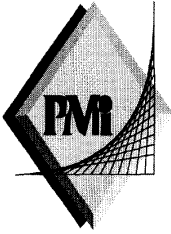
The auditor's agency name, address, phone number, and Attestation letter with a brief audit summary report indicating compliance must be included in the full disclosure report. A statement should be included specifying who to contact in order to obtain further information regarding the audit process.

This implementation of the TPC Benchmark D was audited by Lorna Livingtree of Performance Metrics, Inc. Further information regarding the audit process may be obtained from:

Performance Metrics, Inc.
2229 Benita Dr. Suite 101
Rancho Cordova, CA
(phone) 916/635-2822
(fax) 916/858-0109

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

Transaction Processing Performance Council
c/o Shanley Public Relations
777 North First Street, Suite 6000
San Jose, CA 95112-6311
408/295-8894



PERFORMANCE METRICS INC.
TPC Certified Auditors

June 19, 1998

Mr. Bryon Georgson
Database Performance Engineer
Compaq Computer Corporation
20555 SH 249
Houston, TX 77070

I have verified on site the TPC Benchmark™ D for the following configuration:

Platform: ProLiant 7000 X/400
Database Manager: Oracle8 Enterprise Edition release 8.0.4
Operating System: Microsoft NT version 4.0

CPU's	Memory	Total Disks	QppD@100GB	QthD@100GB	QphD@100GB
4 Xeon @ 400 MHz	Main: 4 GB	84 @ 9.1 GB 1 @ 4.3 GB	1,694.6	553.1	968.1

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 100GB using DBGEN with modifications. The TIME table was not defined or used.
- The qualification database layout was identical to the tested database except for number of rows.
- The query text was verified to use only compliant variants and minor modifications.
- The executable query text was generated by QGEN and submitted to Oracle through a compliant implementation specific layer.
- The validation of the query text against the qualification database produced compliant results.
- The update functions were properly implemented and executed.
- The load timing was properly measured and reported.
- The execution times were correctly measured.

PERFORMANCE METRICS INC.
TPC Certified Auditors

- The performance metrics were correctly computed.
- The throughput metric was computed using the results of the power test.
- The repeatability of the measurement was verified.
- The ACID properties were tested and verified.
- Eight hours of 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:
None.

Sincerely,



Lorna Livingtree
Auditor

Appendix A: Parameter Settings

RUN.ORA

```
O7_DICTIONARY_ACCESSIBILITY = TRUE
allow_partial_sn_results = FALSE
always_anti_join = HASH
aq_tm_processes = 0
arch_io_slaves = 0
audit_trail = FALSE
b_tree_bitmap_plans = FALSE
background_dump_dest = c:\tpcd\log\hundgig
backup_disk_io_slaves = 0
backup_tape_io_slaves = FALSE
bitmap_merge_area_size = 1048576
blank_trimming = FALSE
#buffer_pool_keep =
#buffer_pool_recycle =
cache_size_threshold = 400
cleanup_rollback_entries = 100
close_cached_open_cursors = FALSE
commit_point_strength = 1
compatible = 8.0.3
compatible_no_recovery = 8.0.0
control_file_record_keep_time = 7
control_files = c:\tpcd\dfs\hundgig\cntr\tpcd.dbf
cpu_count = 4
create_bitmap_area_size = 8388608
cursor_space_for_time = FALSE
db_block_buffers = 4000
db_block_checkpoint_batch = 8
db_block_checksum = FALSE
db_block_lru_extended_statistics = 0
db_block_lru_latches = 24
db_block_lru_statistics = FALSE
db_block_max_dirty_target = 4294967294
db_block_size = 16384
db_domain = WORLD
db_file_direct_io_count = 64
db_file_multiblock_read_count = 32
#db_file_name_convert =
db_file_simultaneous_writes = 4
db_files = 10000
db_name = tpcd
dblink_encrypt_login = FALSE
dbwr_io_slaves = 0
delayed_logging_block_cleanouts = TRUE
discrete_transactions_enabled = FALSE
disk_asynch_io = TRUE
distributed_lock_timeout = 60
#distributed_recovery_connection_hol = 200
distributed_transactions = 0
dml_locks = 2000
enqueue_resources = 15000
#event =
#fixed_date =
#freeze_DB_for_fast_instance_recover = FALSE
gc_defer_time = 10
#gc_files_to_locks =
gc_ckpt_procs = 1
gc_releasable_locks = 0
#gc_rollback_locks =
global_names = FALSE
hash_area_size = 10485760
hash_join_enabled = TRUE
hash_multiblock_io_count = 8
#instance_groups =
instance_number = 0
job_queue_interval = 60
job_queue_keep_connections = FALSE
job_queue_processes = 0
large_pool_min_alloc = 16K
large_pool_size = 0
lgwr_io_slaves = 0
license_max_sessions = 0
license_max_users = 0
license_sessions_warning = 0
lm_locks = 12000
lm_procs = 127
lm_ress = 6000
#local_listener =
#lock_name_space =
log_archive_buffer_size = 127
log_archive_buffers = 4
#log_archive_dest = %RDBMS80\
#log_archive_duplex_dest =
#log_archive_format = ARC%s.%t
log_archive_min_succeed_dest = 1
log_archive_start = FALSE
log_block_checksum = FALSE
log_buffer = 5242880
log_checkpoint_interval = 2000000000
log_checkpoint_timeout = 0
log_checkpoints_to_alert = TRUE
#log_file_name_convert =
log_files = 255
log_simultaneous_copies = 4
log_small_entry_max_size = 80
max_commit_propagation_delay = 90000
max_dump_file_size = 4000000
max_enabled_roles = 20
max_rollback_segments = 256
max_transaction_branches = 8
#mts_dispatcher =
#mts_listener_address =
mts_max_dispatchers = 5
mts_max_servers = 20
mts_multiple_listeners = FALSE
#mts_rate_log_size =
#mts_rate_scale =
mts_servers = 0
mts_service = tpcd
#nls_currency =
nls_date_format = YYYY-MM-DD
#nls_date_language =
#nls_iso_currency =
nls_language = AMERICAN
#nls_numeric_characters =
#nls_sort =
nls_territory = AMERICA
object_cache_max_size_percent = 10
object_cache_optimal_size = 102400
open_cursors = 1024
open_links = 4
open_links_per_instance = 4
#ops_admin_group =
optimizer_mode = CHOOSE
```

```

optimizer_percent_parallel      = 100
optimizer_search_limit          = 6
#oracle_trace_collection_name   =
#oracle_trace_collection_path   = %OTRACE80%ADMIN\CDF\
oracle_trace_collection_size     = 5242880
oracle_trace_enable             = FALSE
oracle_trace_facility_name       = oracled
#oracle_trace_facility_path      = %OTRACE80%ADMIN\FDF\
os_authent_prefix               = OPS$
os_roles                        = FALSE
parallel_default_max_instances  = 0
#parallel_instance_group        =
parallel_max_servers            = 60
parallel_min_message_pool       = 1546560
parallel_min_percent            = 0
parallel_min_servers            = 60
parallel_server                 = FALSE
parallel_server_idle_time       = 5
parallel_transaction_resource_timeout = 300
partition_view_enabled          = FALSE
plsql_v2_compatibility          = FALSE
pre_page_sga                   = FALSE
processes                      = 240
recovery_parallelism            = 0
remote_dependencies_mode        = TIMESTAMP
remote_login_passwordfile       = SHARED
remote_os_authent               = FALSE
remote_os_roles                 = FALSE
replication_dependency_tracking = TRUE
resource_limit                  = FALSE
rollback_segments               = r1, r2, r3, r4, r5, r6, r7, r8
row_cache_cursors               = 10
row_locking                    = always
sequence_cache_entries          = 10
sequence_cache_hash_buckets     = 7
serial_reuse                    = DISABLE
serializable                   = FALSE
session_cached_cursors          = 0
session_max_open_files          = 0
sessions                      = 512
shared_pool_reserved_min_alloc  = 5K
shared_pool_reserved_size       = 4500000
shared_pool_size                = 90000000
snapshot_refresh_interval       = 60
snapshot_refresh_keep_connections = FALSE
snapshot_refresh_processes      = 0
sort_area_retained_size         = 1048576
sort_area_size                  = 1048576
sort_direct_writes              = TRUE
sort_read_fac                   = 15
sort_spacemap_size              = 512
sort_write_buffer_size          = 65536
sort_write_buffers              = 8
spin_count                     = 2000
spread_extents                  = TRUE
sql92_security                  = FALSE
sql_trace                      = FALSE
star_transformation_enabled     = FALSE
tape_asynch_io                  = TRUE
temporary_table_locks          = 256
text_enable                     = FALSE
thread                         = 1
timed_os_statistics             = 0
timed_statistics                = FALSE
transaction_auditing            = TRUE
transactions                    = 512
transactions_per_rollback_segment = 1
user_dump_dest                  = c:\tpcd\log\hundgig

```

```
#utl_file_dir =
```

SUSAN.ORA

```

O7_DICTIONARY_ACCESSIBILITY    = TRUE
_advanced_dss_features          = TRUE
_db_block_write_batch           = 1024
allow_partial_sn_results        = FALSE
always_anti_join                = HASH
always_semi_join                = HASH
aq_tm_processes                 = 0
arch_io_slaves                  = 0
audit_trail                     = FALSE
b_tree_bitmap_plans             = FALSE
background_dump_dest            = c:\tpcd\log\hundgig
backup_disk_io_slaves           = 0
backup_tape_io_slaves           = FALSE
bitmap_merge_area_size          = 1048576
blank_trimming                  = FALSE
#buffer_pool_keep               =
#buffer_pool_recycle            =
cache_size_threshold            = 800
cleanup_rollback_entries        = 1000
close_cached_open_cursors       = FALSE
commit_point_strength           = 1
compatible                      = 8.0.4
compatible_no_recovery          = 8.0.0
complex_view_merging            = TRUE
control_file_record_keep_time   = 7
control_files                   = c:\tpcd\dfs\hundgig\cntrltpcd.dbf
cpu_count                       = 4
create_bitmap_area_size         = 8388608
cursor_space_for_time           = FALSE
db_block_buffers                = 8000
db_block_checkpoint_batch       = 8
db_block_checksum               = FALSE
db_block_lru_extended_statistics = 0
db_block_lru_latches            = 24
db_block_lru_statistics         = FALSE
db_block_max_dirty_target       = 4294967294
db_block_size                   = 16384
db_domain                       = WORLD
db_file_direct_io_count         = 64
db_file_multiblock_read_count   = 8
#db_file_name_convert           =
db_file_simultaneous_writes     = 20
db_files                        = 3000
db_name                         = tpcd
dblink_encrypt_login            = FALSE
dbwr_io_slaves                  = 0
delayed_logging_block_cleanouts = TRUE
discrete_transactions_enabled    = FALSE
disk_asynch_io                  = TRUE
distributed_lock_timeout        = 60
#distributed_recovery_connection_hold = 200
distributed_transactions        = 0
dml_locks                       = 2000
enqueue_resources               = 15000
fast_full_scan_enabled          = TRUE
gc_defer_time                   = 10
gc_lck_procs                    = 1
gc_releasable_locks            = 0
global_names                    = FALSE
#hash_area_size                 = 20971520
#hash_area_size                 = 20000000
#hash_area_size                 = 12000000
#hash_area_size                 = 10485760

```

hash_area_size	= 10000000	#oracle_trace_collection_path	= %OTRACE80%\ADMIN\CDF\
hash_join_enabled	= TRUE	oracle_trace_collection_size	= 5242880
hash_multiblock_io_count	= 16	oracle_trace_enable	= FALSE
#instance_groups	=	oracle_trace_facility_name	= oracled
instance_number	= 0	ordered_nested_loop	= TRUE
job_queue_interval	= 60	#oracle_trace_facility_path	= %OTRACE80%\ADMIN\FDF\
job_queue_keep_connections	= FALSE	os_authent_prefix	= OP\$S
job_queue_processes	= 0	os_roles	= FALSE
large_pool_min_alloc	= 16K	parallel_default_max_instances	= 0
large_pool_size	= 0	parallel_execution_message_size	= 20480
lgwr_io_slaves	= 0	#parallel_instance_group	=
license_max_sessions	= 0	parallel_max_servers	= 80
license_max_users	= 0	parallel_min_message_pool	= 80000000
license_sessions_warning	= 0	#parallel_min_message_pool	= 1546560
lm_locks	= 12000	parallel_min_percent	= 0
lm_procs	= 127	parallel_min_servers	= 80
lm_res	= 6000	parallel_server	= FALSE
#local_listener	=	parallel_server_idle_time	= 5
#lock_name_space	=	parallel_transaction_resource_timeout	= 300
log_archive_buffer_size	= 127	partition_view_enabled	= FALSE
log_archive_buffers	= 4	plsql_v2_compatibility	= FALSE
#log_archive_dest	= %RDBMS80%\	pre_page_sga	= FALSE
#log_archive_duplex_dest	=	processes	= 150
#log_archive_format	= ARC%s.%t	push_join_predicate	= TRUE
log_archive_min_succeed_dest	= 1	recovery_parallelism	= 0
log_archive_start	= FALSE	remote_dependencies_mode	= TIMESTAMP
log_block_checksum	= FALSE	remote_login_passwordfile	= SHARED
log_buffer	= 5242880	remote_os_authent	= FALSE
log_checkpoint_interval	= 2000000000	remote_os_roles	= FALSE
log_checkpoint_timeout	= 0	replication_dependency_tracking	= TRUE
log_checkpoints_to_alert	= TRUE	resource_limit	= FALSE
#log_file_name_convert	=	rollback_segments	=
log_files	= 255	r1,r2,r3,r4,r5,r6,r7,r8,r9,r10,r11,r12,r13,r14,r15,r16,r17,r18,r19,r20,r21,r22,r23,r24,r25,r26,r27,r28,r29,r30,r31,r32,r33,r34,r35,r36,r37,r38,r39,r40	
log_simultaneous_copies	= 4	row_cache_cursors	= 10
log_small_entry_max_size	= 80	row_locking	= always
max_commit_propagation_delay	= 90000	sequence_cache_entries	= 10
max_dump_file_size	= 4000000	sequence_cache_hash_buckets	= 7
max_enabled_roles	= 20	serial_reuse	= DISABLE
max_rollback_segments	= 256	serializable	= FALSE
max_transaction_branches	= 8	session_cached_cursors	= 0
#mts_dispatchers	=	session_max_open_files	= 0
#mts_listener_address	=	sessions	= 256
mts_max_dispatchers	= 5	shared_pool_reserved_min_alloc	= 5K
mts_max_servers	= 20	shared_pool_reserved_size	= 6500000
mts_multiple_listeners	= FALSE	#shared_pool_reserved_size	= 2000000
#mts_rate_log_size	=	#shared_pool_size	= 40000000
#mts_rate_scale	=	shared_pool_size	= 130000000
mts_servers	= 0	snapshot_refresh_interval	= 60
mts_service	= tpcd	snapshot_refresh_keep_connections	= FALSE
#nls_currency	=	snapshot_refresh_processes	= 0
nls_date_format	= YYYY-MM-DD	sort_area_retained_size	= 1048576
#nls_date_language	=	sort_area_size	= 1048576
#nls_iso_currency	=	sort_direct_writes	= AUTO
nls_language	= AMERICAN	sort_read_fac	= 15
#nls_numeric_characters	=	sort_spacemap_size	= 512
#nls_sort	=	sort_write_buffer_size	= 65536
nls_territory	= AMERICA	sort_write_buffers	= 20
object_cache_max_size_percent	= 10	spin_count	= 2000
object_cache_optimal_size	= 102400	spread_extents	= TRUE
open_cursors	= 120	sql92_security	= FALSE
open_links	= 4	sql_trace	= FALSE
open_links_per_instance	= 4	star_transformation_enabled	= FALSE
#ops_admin_group	=	tape_asynch_io	= TRUE
optimizer_mode	= CHOOSE	temporary_table_locks	= 256
optimizer_percent_parallel	= 100	text_enable	= FALSE
optimizer_search_limit	= 6		
#oracle_trace_collection_name	=		

```
thread = 0
#thread = 1
timed_os_statistics = 0
timed_statistics = FALSE
transaction_auditing = TRUE
transactions = 512
transactions_per_rollback_segment = 2
user_dump_dest = c:\tpcd\log\hundgig
#utl_file_dir =
```

NT Registry Changes

Key Name:

SYSTEM\CurrentControlSet\Control\Class\{4D36E97B-E325-11CE-BFC1-08002BE10318}\0004

Class Name: <NO CLASS>

Last Write Time: 6/3/98 - 3:33 PM

Name: DriverDesc
Type: REG_SZ
Data: Compaq SMART-2 Array Controllers ECI Driver

Key Name:

SOFTWARE\ORACLE

Class Name: <NO CLASS>

Last Write Time: 6/3/98 - 3:33 PM

Name: DBA_AUTHORIZATION
Type: REG_SZ
Data: manager

Name: ORA_TPCD_PWFILE
Type: REG_EXPAND_SZ
Data: C:\ORANT\DATABASE\PWDTPCD.ORA

Name: ORA_WORKINGSETMAX
Type: REG_DWORD
Data: 0xb54

Name: ORA_WORKINGSETMIN
Type: REG_DWORD
Data: 0xaf0

Name: ORACLE_PRIORITY
Type: REG_EXPAND_SZ
Data: CLASS:high;DEF:7

Name: ORACLE_SID
Type: REG_SZ
Data: TPCD

```
# Filename..... listener.ora
# Node..... local.world
# Date..... 24-MAY-94 13:23:20
#####
```

```
LISTENER =
  (ADDRESS_LIST =
    (ADDRESS=
      (PROTOCOL= IPC)
      (KEY= oracle.world)
    )
    (ADDRESS=
      (PROTOCOL= IPC)
      (KEY= TPCD)
    )
    (ADDRESS=
      (COMMUNITY= NMP.world)
      (PROTOCOL= NMP)
      (SERVER= HOTDOG)
      (PIPE= ORAPIPE)
    )
    (ADDRESS=
      (COMMUNITY= TCP.world)
      (PROTOCOL= TCP)
      (Host= HOTDOG)
      (Port= 1521)
    )
    (ADDRESS=
      (COMMUNITY= TCP.world)
      (PROTOCOL= TCP)
      (Host= HOTDOG)
      (Port= 1526)
    )
    (ADDRESS=
      (COMMUNITY= TCP.world)
      (PROTOCOL= TCP)
      (Host= 127.0.0.1)
      (Port= 1521)
    )
  )
STARTUP_WAIT_TIME_LISTENER = 0
CONNECT_TIMEOUT_LISTENER = 10
TRACE_LEVEL_LISTENER = ADMIN
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = TPCD)
    )
  )
PASSWORDS_LISTENER = (oracle)
```

TNSNAMES.ORA

```
tpcd.world =
  (DESCRIPTION=(address_list=
    (ADDRESS=
      (PROTOCOL=tpc)(PORT=1610)(HOST=CM_SERVER))
    (ADDRESS=
      (PROTOCOL=tpc)(PORT=1521)(HOST=LSNR_SERVER)))
    (CONNECT_DATA=(SID=TPCD))
    (SOURCE_ROUTE=yes)
  )
)
```

LISTENER.ORA

```
#####
```

Appendix B: Table Definitions

BUILDIT.SH

```
#####
#####
```

```
# Database Creation Phase
# creating database and initial rollback segment
svrmgr30 <<!
connect internal/manager
startup pfile=c:\tpcd\tdbs\hundredgig\run.ora nomount;
create database
  controlfile reuse
  logfile '\\\\.\\ORA_LOG1' size 1999m reuse,
  '\\\\.\\ORA_LOG2' size 1999m reuse
  datafile '\\\\.\\ORA_SYS1' size 999m reuse
  maxdatafiles 10000
;
```

```
create public rollback segment t_rs1 storage (initial 1024k next 1024k
maxextents unlimited);
```

```
alter rollback segment t_rs1 online;
```

```
shutdown
startup pfile=c:\tpcd\tdbs\hundredgig\run.ora;
create tablespace ts_undo
  datafile '\\\\.\\ORA_ROLL1' size 3499m reuse;
create public rollback segment r1 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r2 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r3 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r4 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r5 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r6 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r7 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r8 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r9 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r10 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r11 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r12 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r13 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r14 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r15 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r16 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
```

```
create public rollback segment r17 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r18 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r19 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r20 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r21 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r22 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r23 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r24 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r25 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r26 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r27 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r28 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r29 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r30 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r31 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r32 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r33 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r34 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r35 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r36 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r37 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r38 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r39 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
create public rollback segment r40 storage (initial 1024k next 1024k
maxextents unlimited) tablespace ts_undo;
set termout off
set echo off
@c:\orant\tdbms80\admin\catalog.sql;
@c:\orant\tdbms80\admin\catparr.sql;
@c:\orant\tdbms80\admin\catproc.sql;
@c:\orant\tdbms80\admin\utlxplan.sql;
shutdown;
startup pfile=c:\tpcd\tdbs\hundredgig\run.ora;
drop tablespace ts_ps including contents;
create tablespace ts_ps
  datafile '\\\\.\\ORA_PS_1' size 299m reuse
  default storage (initial 64k next 290m pctincrease 0)
  nologging;
drop tablespace ts_s including contents;
create tablespace ts_s
  datafile '\\\\.\\ORA_SUPPLIER_1' size 49m reuse
  default storage (initial 64k next 40m pctincrease 0)
  nologging;
```



```

:
drop tablespace ts_i69 including contents;
create tablespace ts_i69
  datafile '\\\\\\WORA_LINEITEM_69' size 949m reuse
  default storage (initial 64k next 880m pctincrease 0)
  nologging

:
drop tablespace ts_i70 including contents;
create tablespace ts_i70
  datafile '\\\\\\WORA_LINEITEM_70' size 949m reuse
  default storage (initial 64k next 880m pctincrease 0)
  nologging

:
drop tablespace ts_i71 including contents;
create tablespace ts_i71
  datafile '\\\\\\WORA_LINEITEM_71' size 949m reuse
  default storage (initial 64k next 880m pctincrease 0)
  nologging

:
drop tablespace ts_i72 including contents;
create tablespace ts_i72
  datafile '\\\\\\WORA_LINEITEM_72' size 949m reuse
  default storage (initial 64k next 880m pctincrease 0)
  nologging

:
drop tablespace ts_i73 including contents;
create tablespace ts_i73
  datafile '\\\\\\WORA_LINEITEM_73' size 949m reuse
  default storage (initial 64k next 880m pctincrease 0)
  nologging

:
drop tablespace ts_i74 including contents;
create tablespace ts_i74
  datafile '\\\\\\WORA_LINEITEM_74' size 949m reuse
  default storage (initial 64k next 880m pctincrease 0)
  nologging

:
drop tablespace ts_i75 including contents;
create tablespace ts_i75
  datafile '\\\\\\WORA_LINEITEM_75' size 949m reuse
  default storage (initial 64k next 880m pctincrease 0)
  nologging

:
drop tablespace ts_i76 including contents;
create tablespace ts_i76
  datafile '\\\\\\WORA_LINEITEM_76' size 949m reuse
  default storage (initial 64k next 880m pctincrease 0)
  nologging

:
drop tablespace ts_i77 including contents;
create tablespace ts_i77
  datafile '\\\\\\WORA_LINEITEM_77' size 949m reuse
  default storage (initial 64k next 880m pctincrease 0)
  nologging

:
drop tablespace ts_i78 including contents;
create tablespace ts_i78
  datafile '\\\\\\WORA_LINEITEM_78' size 949m reuse
  default storage (initial 64k next 880m pctincrease 0)
  nologging

:
drop tablespace ts_i79 including contents;
create tablespace ts_i79
  datafile '\\\\\\WORA_LINEITEM_79' size 949m reuse
  default storage (initial 64k next 880m pctincrease 0)
  nologging

:

```

```

drop tablespace ts_i80 including contents;
create tablespace ts_i80
  datafile '\\\\\\WORA_LINEITEM_80' size 949m reuse
  default storage (initial 64k next 880m pctincrease 0)
  nologging

:
drop tablespace ts_o1 including contents;
create tablespace ts_o1
  datafile '\\\\\\WORA_ORDER_1' size 219m reuse
  default storage (initial 64k next 210m pctincrease 0)
  nologging

:
drop tablespace ts_o2 including contents;
create tablespace ts_o2
  datafile '\\\\\\WORA_ORDER_2' size 219m reuse
  default storage (initial 64k next 210m pctincrease 0)
  nologging

:
drop tablespace ts_o3 including contents;
create tablespace ts_o3
  datafile '\\\\\\WORA_ORDER_3' size 219m reuse
  default storage (initial 64k next 210m pctincrease 0)
  nologging

:
drop tablespace ts_o4 including contents;
create tablespace ts_o4
  datafile '\\\\\\WORA_ORDER_4' size 219m reuse
  default storage (initial 64k next 210m pctincrease 0)
  nologging

:
drop tablespace ts_o5 including contents;
create tablespace ts_o5
  datafile '\\\\\\WORA_ORDER_5' size 219m reuse
  default storage (initial 64k next 210m pctincrease 0)
  nologging

:
drop tablespace ts_o6 including contents;
create tablespace ts_o6
  datafile '\\\\\\WORA_ORDER_6' size 219m reuse
  default storage (initial 64k next 210m pctincrease 0)
  nologging

:
drop tablespace ts_o7 including contents;
create tablespace ts_o7
  datafile '\\\\\\WORA_ORDER_7' size 219m reuse
  default storage (initial 64k next 210m pctincrease 0)
  nologging

:
drop tablespace ts_o8 including contents;
create tablespace ts_o8
  datafile '\\\\\\WORA_ORDER_8' size 219m reuse
  default storage (initial 64k next 210m pctincrease 0)
  nologging

:
drop tablespace ts_o9 including contents;
create tablespace ts_o9
  datafile '\\\\\\WORA_ORDER_9' size 219m reuse
  default storage (initial 64k next 210m pctincrease 0)
  nologging

:
drop tablespace ts_o10 including contents;
create tablespace ts_o10
  datafile '\\\\\\WORA_ORDER_10' size 219m reuse
  default storage (initial 64k next 210m pctincrease 0)
  nologging

:
drop tablespace ts_o11 including contents;

```



```
drop tablespace ts_o44 including contents;
create tablespace ts_o44
datafile '1\1\ORA_ORDER_44' size 219m reuse
default storage (initial 64k next 210m pctincrease 0)
nologging
```

```
drop tablespace ts_o56 including contents;
```



```
drop tablespace ts_o79 including contents;
create tablespace ts_o79
  datafile '\\\\\\\\ORA_ORDER_79' size 219m reuse
  default storage (initial 64k next 210m pctincrease 0)
  nologging
```

```
drop tablespace ts_o80 including contents;
create tablespace ts_o80
  datafile '\\\\\\\\ORA_ORDER_80' size 219m reuse
  default storage (initial 64k next 210m pctincrease 0)
  nologging
```

```
drop tablespace ts_c1 including contents;
create tablespace ts_c1
  datafile '\\\\\\\\ORA_CUSTOMER1_1' size 15m reuse
  default storage (initial 64k next 7m pctincrease 0)
  nologging
```

```
drop tablespace ts_c2 including contents;
create tablespace ts_c2
  datafile '\\\\\\\\ORA_CUSTOMER2_1' size 15m reuse
  default storage (initial 64k next 7m pctincrease 0)
  nologging
```

```
drop tablespace ts_c3 including contents;
create tablespace ts_c3
  datafile '\\\\\\\\ORA_CUSTOMER3_1' size 15m reuse
  default storage (initial 64k next 7m pctincrease 0)
  nologging
```

```
drop tablespace ts_c4 including contents;
create tablespace ts_c4
  datafile '\\\\\\\\ORA_CUSTOMER4_1' size 15m reuse
  default storage (initial 64k next 7m pctincrease 0)
  nologging
```

```
drop tablespace ts_c5 including contents;
create tablespace ts_c5
  datafile '\\\\\\\\ORA_CUSTOMER5_1' size 15m reuse
  default storage (initial 64k next 7m pctincrease 0)
  nologging
```

```
drop tablespace ts_p1 including contents;
create tablespace ts_p1
  datafile '\\\\\\\\ORA_PARTS_1' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging
```

```
drop tablespace ts_p2 including contents;
create tablespace ts_p2
  datafile '\\\\\\\\ORA_PARTS_2' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging
```

```
drop tablespace ts_p3 including contents;
create tablespace ts_p3
  datafile '\\\\\\\\ORA_PARTS_3' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging
```

```
drop tablespace ts_p4 including contents;
create tablespace ts_p4
  datafile '\\\\\\\\ORA_PARTS_4' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging
```

```
drop tablespace ts_p5 including contents;
create tablespace ts_p5
  datafile '\\\\\\\\ORA_PARTS_5' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging
```

```
drop tablespace ts_p6 including contents;
create tablespace ts_p6
  datafile '\\\\\\\\ORA_PARTS_6' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging
```

```
drop tablespace ts_p7 including contents;
create tablespace ts_p7
  datafile '\\\\\\\\ORA_PARTS_7' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging
```

```
drop tablespace ts_p8 including contents;
create tablespace ts_p8
  datafile '\\\\\\\\ORA_PARTS_8' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging
```

```
drop tablespace ts_p9 including contents;
create tablespace ts_p9
  datafile '\\\\\\\\ORA_PARTS_9' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging
```

```
drop tablespace ts_p10 including contents;
create tablespace ts_p10
  datafile '\\\\\\\\ORA_PARTS_10' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging
```

```
drop tablespace ts_p11 including contents;
create tablespace ts_p11
  datafile '\\\\\\\\ORA_PARTS_11' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging
```

```
drop tablespace ts_p12 including contents;
create tablespace ts_p12
  datafile '\\\\\\\\ORA_PARTS_12' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging
```

```
drop tablespace ts_p13 including contents;
create tablespace ts_p13
  datafile '\\\\\\\\ORA_PARTS_13' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging
```

```
drop tablespace ts_p14 including contents;
create tablespace ts_p14
  datafile '\\\\\\\\ORA_PARTS_14' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging
```

```
drop tablespace ts_p15 including contents;
create tablespace ts_p15
  datafile '\\\\\\\\ORA_PARTS_15' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging
```

```
drop tablespace ts_p16 including contents;
```



```

drop tablespace ts_p39 including contents;
create tablespace ts_p39
  datafile '\\\\\\\\ORA_PARTS_39' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging

```

```

drop tablespace ts_p40 including contents;
create tablespace ts_p40
  datafile '\\\\\\\\ORA_PARTS_40' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging

```

```

drop tablespace ts_p41 including contents;
create tablespace ts_p41
  datafile '\\\\\\\\ORA_PARTS_41' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging

```

```

drop tablespace ts_p42 including contents;
create tablespace ts_p42
  datafile '\\\\\\\\ORA_PARTS_42' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging

```

```

drop tablespace ts_p43 including contents;
create tablespace ts_p43
  datafile '\\\\\\\\ORA_PARTS_43' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging

```

```

drop tablespace ts_p44 including contents;
create tablespace ts_p44
  datafile '\\\\\\\\ORA_PARTS_44' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging

```

```

drop tablespace ts_p45 including contents;
create tablespace ts_p45
  datafile '\\\\\\\\ORA_PARTS_45' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging

```

```

drop tablespace ts_p46 including contents;
create tablespace ts_p46
  datafile '\\\\\\\\ORA_PARTS_46' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging

```

```

drop tablespace ts_p47 including contents;
create tablespace ts_p47
  datafile '\\\\\\\\ORA_PARTS_47' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging

```

```

drop tablespace ts_p48 including contents;
create tablespace ts_p48
  datafile '\\\\\\\\ORA_PARTS_48' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging

```

```

drop tablespace ts_p49 including contents;
create tablespace ts_p49
  datafile '\\\\\\\\ORA_PARTS_49' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging

```

```

drop tablespace ts_p50 including contents;
create tablespace ts_p50
  datafile '\\\\\\\\ORA_PARTS_50' size 62m reuse
  default storage (initial 64k next 2m pctincrease 0)
  nologging

```

```

drop tablespace ts_ind1 including contents;
create tablespace ts_ind1
  datafile '\\\\\\\\ORA_IND1_1' size 1199m reuse
  default storage (initial 10m next 10m maxextents unlimited
  pctincrease 0)
  nologging

```

```

drop tablespace ts_ind3 including contents;
create tablespace ts_ind3
  datafile '\\\\\\\\ORA_IND3_1' size 299m reuse
  default storage (initial 7m next 7m maxextents unlimited pctincrease
  0)
  nologging

```

```

drop tablespace ts_ind4 including contents;
create tablespace ts_ind4
  datafile '\\\\\\\\ORA_IND4_1' size 277m reuse
  default storage (initial 7m next 7m maxextents unlimited pctincrease
  0)
  nologging

```

```

drop tablespace ts_ind5 including contents;
create tablespace ts_ind5
  datafile '\\\\\\\\ORA_IND5_1' size 1999m reuse
  default storage (initial 10m next 10m maxextents unlimited
  pctincrease 0)
  nologging

```

```

drop tablespace ts_temp including contents;
create tablespace ts_temp temporary
  datafile '\\\\\\\\ORA_TEMP_1' size 1989m reuse
  default storage (initial 140m next 140m maxextents unlimited
  pctincrease 0)
  nologging

```

```

!
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_2' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_3' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_4' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_5' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_6' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_7' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_8' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_9' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_10' 299m &
wait
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_11' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_12' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_13' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_14' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_15' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_16' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_17' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_18' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_19' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_20' 299m &
wait
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_21' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_22' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_23' 299m &
plus80 system/manager @addfile ts_ps '\\\\\\ORA_PS_24' 299m &

```



```
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_15'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_16'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_17'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_18'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_19'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_20'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_21'  
15m &  
wait  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_22'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_23'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_24'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_25'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_26'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_27'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_28'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_29'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_30'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_31'  
15m &  
wait  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_32'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_33'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_34'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_35'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_36'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_37'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_38'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_39'  
15m &  
plus80 system/manager @addfile ts_c4 "\\.\ORA_CUSTOMER4_40'  
15m &  
wait  
plus80 system/manager @addfile ts_c5 "\\.\ORA_CUSTOMER5_2'  
15m &  
plus80 system/manager @addfile ts_c5 "\\.\ORA_CUSTOMER5_3'  
15m &  
plus80 system/manager @addfile ts_c5 "\\.\ORA_CUSTOMER5_4'  
15m &  
plus80 system/manager @addfile ts_c5 "\\.\ORA_CUSTOMER5_5'  
15m &  
plus80 system/manager @addfile ts_c5 "\\.\ORA_CUSTOMER5_6'  
15m &  
plus80 system/manager @addfile ts_c5 "\\.\ORA_CUSTOMER5_7'
```


[illegible][illegible]

```

plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_17' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_18' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_19' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_20' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_21' 1989m
&
wait
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_22' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_23' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_24' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_25' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_26' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_27' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_28' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_29' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_30' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_31' 1989m
&
wait
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_32' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_33' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_34' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_35' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_36' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_37' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_38' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_39' 1989m
&
plus80 system/manager @addfile ts_temp '\\.ORA_TEMP_40' 1989m
&
wait

```

LOADIT.SH

```

echo "dropping old information at `date`"
svrmgr30 <<!
connect internal/manager
select to_char(sysdate, 'MM/DD/YYYY HH24:MI:SS') from dual;
drop user tpcd cascade;

```

```

alter tablespace ts_undo coalesce;
alter tablespace ts_ps coalesce;
alter tablespace ts_s coalesce;

```

```

alter tablespace ts_i1 coalesce;
alter tablespace ts_i2 coalesce;
alter tablespace ts_i3 coalesce;
alter tablespace ts_i4 coalesce;
alter tablespace ts_i5 coalesce;
alter tablespace ts_i6 coalesce;
alter tablespace ts_i7 coalesce;
alter tablespace ts_i8 coalesce;
alter tablespace ts_i9 coalesce;
alter tablespace ts_i10 coalesce;
alter tablespace ts_i11 coalesce;
alter tablespace ts_i12 coalesce;
alter tablespace ts_i13 coalesce;
alter tablespace ts_i14 coalesce;
alter tablespace ts_i15 coalesce;
alter tablespace ts_i16 coalesce;
alter tablespace ts_i17 coalesce;
alter tablespace ts_i18 coalesce;
alter tablespace ts_i19 coalesce;
alter tablespace ts_i20 coalesce;
alter tablespace ts_i21 coalesce;
alter tablespace ts_i22 coalesce;
alter tablespace ts_i23 coalesce;
alter tablespace ts_i24 coalesce;
alter tablespace ts_i25 coalesce;
alter tablespace ts_i26 coalesce;
alter tablespace ts_i27 coalesce;
alter tablespace ts_i28 coalesce;
alter tablespace ts_i29 coalesce;
alter tablespace ts_i30 coalesce;
alter tablespace ts_i31 coalesce;
alter tablespace ts_i32 coalesce;
alter tablespace ts_i33 coalesce;
alter tablespace ts_i34 coalesce;
alter tablespace ts_i35 coalesce;
alter tablespace ts_i36 coalesce;
alter tablespace ts_i37 coalesce;
alter tablespace ts_i38 coalesce;
alter tablespace ts_i39 coalesce;
alter tablespace ts_i40 coalesce;
alter tablespace ts_i41 coalesce;
alter tablespace ts_i42 coalesce;
alter tablespace ts_i43 coalesce;
alter tablespace ts_i44 coalesce;
alter tablespace ts_i45 coalesce;
alter tablespace ts_i46 coalesce;
alter tablespace ts_i47 coalesce;
alter tablespace ts_i48 coalesce;
alter tablespace ts_i49 coalesce;
alter tablespace ts_i50 coalesce;
alter tablespace ts_i51 coalesce;
alter tablespace ts_i52 coalesce;
alter tablespace ts_i53 coalesce;
alter tablespace ts_i54 coalesce;
alter tablespace ts_i55 coalesce;
alter tablespace ts_i56 coalesce;
alter tablespace ts_i57 coalesce;
alter tablespace ts_i58 coalesce;
alter tablespace ts_i59 coalesce;
alter tablespace ts_i60 coalesce;
alter tablespace ts_i61 coalesce;
alter tablespace ts_i62 coalesce;
alter tablespace ts_i63 coalesce;
alter tablespace ts_i64 coalesce;
alter tablespace ts_i65 coalesce;
alter tablespace ts_i66 coalesce;
alter tablespace ts_i67 coalesce;

```

[illegible][illegible]

```

alter tablespace ts_p39 coalesce;
alter tablespace ts_p40 coalesce;
alter tablespace ts_p41 coalesce;
alter tablespace ts_p42 coalesce;
alter tablespace ts_p43 coalesce;
alter tablespace ts_p44 coalesce;
alter tablespace ts_p45 coalesce;
alter tablespace ts_p46 coalesce;
alter tablespace ts_p47 coalesce;
alter tablespace ts_p48 coalesce;
alter tablespace ts_p49 coalesce;
alter tablespace ts_p50 coalesce;
alter tablespace ts_temp coalesce;
alter tablespace ts_ind1 coalesce;
alter tablespace ts_ind3 coalesce;
alter tablespace ts_ind4 coalesce;
alter tablespace ts_ind5 coalesce;

!
echo "Starting scripts to create user and user tables at 'date'"
gtime
svrmgr30 <<!
connect internal/manager;
set termout on;
set echo on;
spool ldevent1.log;
select to_char(sysdate, 'MM/DD/YYYY HH24:MI:SS') from dual;
grant resource,unlimited tablespace,connect
  to tpcd identified by tpcd;
select to_char(sysdate, 'MM/DD/YYYY HH24:MI:SS') from dual;
connect tpcd/tpcd;

drop table lineitem;
create table lineitem (
  l_shipdate      date ,
  l_orderkey      number NOT NULL,
  l_discount      number ,
  l_extendedprice number ,
  l_suppkey       number NOT NULL,
  l_quantity      number ,
  l_returnflag    char(1) ,
  l_partkey       number NOT NULL,
  l_linestatus    char(1) ,
  l_tax           number NOT NULL,
  l_commitdate    date ,
  l_receiptdate   date ,
  l_shipmode      varchar(10) ,
  l_linenum       number NOT NULL,
  l_shipinstruct  varchar(25) ,
  l_comment       varchar(44)
)
pctfree 1
pctused 99
initrans 10
tablespace ts_l1
storage (initial 64k next 895m freelists 16 freelist groups 5 pctincrease
0)
parallel (degree 60 )
partition by range (l_shipdate)
(
partition item1 values less than (to_date('1992-03-24','YYYY-MM-DD'))
tablespace ts_l1

partition item2 values less than (to_date('1992-05-01','YYYY-MM-DD'))
tablespace ts_l2

partition item3 values less than (to_date('1992-06-01','YYYY-MM-DD'))

```

```

tablespace ts_l3

partition item4 values less than (to_date('1992-07-01','YYYY-MM-DD'))
tablespace ts_l4

partition item5 values less than (to_date('1992-08-01','YYYY-MM-DD'))
tablespace ts_l5

partition item6 values less than (to_date('1992-09-01','YYYY-MM-DD'))
tablespace ts_l6

partition item7 values less than (to_date('1992-10-01','YYYY-MM-DD'))
tablespace ts_l7

partition item8 values less than (to_date('1992-11-01','YYYY-MM-DD'))
tablespace ts_l8

partition item9 values less than (to_date('1992-12-01','YYYY-MM-DD'))
tablespace ts_l9

partition item10 values less than (to_date('1993-01-01','YYYY-MM-DD'))
tablespace ts_l10

partition item11 values less than (to_date('1993-02-01','YYYY-MM-DD'))
tablespace ts_l11

partition item12 values less than (to_date('1993-03-01','YYYY-MM-DD'))
tablespace ts_l12

partition item13 values less than (to_date('1993-04-01','YYYY-MM-DD'))
tablespace ts_l13

partition item14 values less than (to_date('1993-05-01','YYYY-MM-DD'))
tablespace ts_l14

partition item15 values less than (to_date('1993-06-01','YYYY-MM-DD'))
tablespace ts_l15

partition item16 values less than (to_date('1993-07-01','YYYY-MM-DD'))
tablespace ts_l16

partition item17 values less than (to_date('1993-08-01','YYYY-MM-DD'))
tablespace ts_l17

partition item18 values less than (to_date('1993-09-01','YYYY-MM-DD'))
tablespace ts_l18

partition item19 values less than (to_date('1993-10-01','YYYY-MM-DD'))
tablespace ts_l19

partition item20 values less than (to_date('1993-11-01','YYYY-MM-DD'))
tablespace ts_l20

partition item21 values less than (to_date('1993-12-01','YYYY-MM-DD'))
tablespace ts_l21

```

```
partition item22 values less than (to_date('1994-01-01','YYYY-MM-DD'))
tablespace ts_j22
```

```
partition item23 values less than (to_date('1994-02-01','YYYY-MM-DD'))
tablespace ts_j23
```

```
partition item24 values less than (to_date('1994-03-01','YYYY-MM-DD'))
tablespace ts_j24
```

```
partition item25 values less than (to_date('1994-04-01','YYYY-MM-DD'))
tablespace ts_j25
```

```
partition item26 values less than (to_date('1994-05-01','YYYY-MM-DD'))
tablespace ts_j26
```

```
partition item27 values less than (to_date('1994-06-01','YYYY-MM-DD'))
tablespace ts_j27
```

```
partition item28 values less than (to_date('1994-07-01','YYYY-MM-DD'))
tablespace ts_j28
```

```
partition item29 values less than (to_date('1994-08-01','YYYY-MM-DD'))
tablespace ts_j29
```

```
partition item30 values less than (to_date('1994-09-01','YYYY-MM-DD'))
tablespace ts_j30
```

```
partition item31 values less than (to_date('1994-10-01','YYYY-MM-DD'))
tablespace ts_j31
```

```
partition item32 values less than (to_date('1994-11-01','YYYY-MM-DD'))
tablespace ts_j32
```

```
partition item33 values less than (to_date('1994-12-01','YYYY-MM-DD'))
tablespace ts_j33
```

```
partition item34 values less than (to_date('1995-01-01','YYYY-MM-DD'))
tablespace ts_j34
```

```
partition item35 values less than (to_date('1995-02-01','YYYY-MM-DD'))
tablespace ts_j35
```

```
partition item36 values less than (to_date('1995-03-01','YYYY-MM-DD'))
tablespace ts_j36
```

```
partition item37 values less than (to_date('1995-04-01','YYYY-MM-DD'))
tablespace ts_j37
```

```
partition item38 values less than (to_date('1995-05-01','YYYY-MM-DD'))
tablespace ts_j38
```

```
partition item39 values less than (to_date('1995-06-01','YYYY-MM-DD'))
tablespace ts_j39
```

```
partition item40 values less than (to_date('1995-07-01','YYYY-MM-DD'))
tablespace ts_j40
```

```
partition item41 values less than (to_date('1995-08-01','YYYY-MM-DD'))
tablespace ts_j41
```

```
partition item42 values less than (to_date('1995-09-01','YYYY-MM-DD'))
tablespace ts_j42
```

```
partition item43 values less than (to_date('1995-10-01','YYYY-MM-DD'))
tablespace ts_j43
```

```
partition item44 values less than (to_date('1995-11-01','YYYY-MM-DD'))
tablespace ts_j44
```

```
partition item45 values less than (to_date('1995-12-01','YYYY-MM-DD'))
tablespace ts_j45
```

```
partition item46 values less than (to_date('1996-01-01','YYYY-MM-DD'))
tablespace ts_j46
```

```
partition item47 values less than (to_date('1996-02-01','YYYY-MM-DD'))
tablespace ts_j47
```

```
partition item48 values less than (to_date('1996-03-01','YYYY-MM-DD'))
tablespace ts_j48
```

```
partition item49 values less than (to_date('1996-04-01','YYYY-MM-DD'))
tablespace ts_j49
```

```
partition item50 values less than (to_date('1996-05-01','YYYY-MM-DD'))
tablespace ts_j50
```

```
partition item51 values less than (to_date('1996-06-01','YYYY-MM-DD'))
tablespace ts_j51
```

```
partition item52 values less than (to_date('1996-07-01','YYYY-MM-DD'))
tablespace ts_j52
```

```
partition item53 values less than (to_date('1996-08-01','YYYY-MM-DD'))
tablespace ts_j53
```

```
partition item54 values less than (to_date('1996-09-01','YYYY-MM-DD'))
tablespace ts_j54
```

```
partition item55 values less than (to_date('1996-10-01','YYYY-MM-DD'))
tablespace ts_j55
```

```

partition item56 values less than (to_date('1996-11-01','YYYY-MM-DD'))
tablespace ts_i56

partition item57 values less than (to_date('1996-12-01','YYYY-MM-DD'))
tablespace ts_i57

partition item58 values less than (to_date('1997-01-01','YYYY-MM-DD'))
tablespace ts_i58

partition item59 values less than (to_date('1997-02-01','YYYY-MM-DD'))
tablespace ts_i59

partition item60 values less than (to_date('1997-03-01','YYYY-MM-DD'))
tablespace ts_i60

partition item61 values less than (to_date('1997-04-01','YYYY-MM-DD'))
tablespace ts_i61

partition item62 values less than (to_date('1997-05-01','YYYY-MM-DD'))
tablespace ts_i62

partition item63 values less than (to_date('1997-06-01','YYYY-MM-DD'))
tablespace ts_i63

partition item64 values less than (to_date('1997-07-01','YYYY-MM-DD'))
tablespace ts_i64

partition item65 values less than (to_date('1997-08-01','YYYY-MM-DD'))
tablespace ts_i65

partition item66 values less than (to_date('1997-09-01','YYYY-MM-DD'))
tablespace ts_i66

partition item67 values less than (to_date('1997-10-01','YYYY-MM-DD'))
tablespace ts_i67

partition item68 values less than (to_date('1997-11-01','YYYY-MM-DD'))
tablespace ts_i68

partition item69 values less than (to_date('1997-12-01','YYYY-MM-DD'))
tablespace ts_i69

partition item70 values less than (to_date('1998-01-01','YYYY-MM-DD'))
tablespace ts_i70

partition item71 values less than (to_date('1998-02-01','YYYY-MM-DD'))
tablespace ts_i71

partition item72 values less than (to_date('1998-03-01','YYYY-MM-DD'))

```

```

tablespace ts_i72

partition item73 values less than (to_date('1998-04-01','YYYY-MM-DD'))
tablespace ts_i73

partition item74 values less than (to_date('1998-05-01','YYYY-MM-DD'))
tablespace ts_i74

partition item75 values less than (to_date('1998-06-01','YYYY-MM-DD'))
tablespace ts_i75

partition item76 values less than (to_date('1998-07-01','YYYY-MM-DD'))
tablespace ts_i76

partition item77 values less than (to_date('1998-08-01','YYYY-MM-DD'))
tablespace ts_i77

partition item78 values less than (to_date('1998-09-01','YYYY-MM-DD'))
tablespace ts_i78

partition item79 values less than (to_date('1998-10-01','YYYY-MM-DD'))
tablespace ts_i79

partition item80 values less than (MAXVALUE)
tablespace ts_i80
)
;

drop table orders;
create table orders (
    o_orderdate    date ,
    o_orderkey     number NOT NULL,
    o_custkey      number NOT NULL,
    o_orderpriority varchar(15) ,
    o_shippriority number ,
    o_clerk         varchar(15) ,
    o_orderstatus  char(1) ,
    o_totalprice   number ,
    o_comment      varchar(79)
)
pctfree 1
pctused 99
initrans 10
tablespace ts_o1
storage (initial 64k next 215m freelists 16 freelist groups 5 pctincrease 0)
parallel (degree 60 )
partition by range (o_orderdate)
(
    partition ord1 values less than (to_date('1992-02-01','YYYY-MM-DD'))
    tablespace ts_o1

    partition ord2 values less than (to_date('1992-03-01','YYYY-MM-DD'))
    tablespace ts_o2

    partition ord3 values less than (to_date('1992-04-01','YYYY-MM-DD'))
    tablespace ts_o3

    partition ord4 values less than (to_date('1992-05-01','YYYY-MM-DD'))
    tablespace ts_o4
)
;

```

```
partition ord28 values less than (to_date('1994-05-01','YYYY-MM-DD'))
tablespace ts_o28

partition ord29 values less than (to_date('1994-06-01','YYYY-MM-DD'))
tablespace ts_o29

partition ord30 values less than (to_date('1994-07-01','YYYY-MM-DD'))
tablespace ts_o30

partition ord31 values less than (to_date('1994-08-01','YYYY-MM-DD'))
tablespace ts_o31

partition ord32 values less than (to_date('1994-09-01','YYYY-MM-DD'))
tablespace ts_o32

partition ord33 values less than (to_date('1994-10-01','YYYY-MM-DD'))
tablespace ts_o33

partition ord34 values less than (to_date('1994-11-01','YYYY-MM-DD'))
tablespace ts_o34

partition ord35 values less than (to_date('1994-12-01','YYYY-MM-DD'))
tablespace ts_o35

partition ord36 values less than (to_date('1995-01-01','YYYY-MM-DD'))
tablespace ts_o36

partition ord37 values less than (to_date('1995-02-01','YYYY-MM-DD'))
tablespace ts_o37

partition ord38 values less than (to_date('1995-03-01','YYYY-MM-DD'))
tablespace ts_o38

partition ord39 values less than (to_date('1995-04-01','YYYY-MM-DD'))
tablespace ts_o39

partition ord40 values less than (to_date('1995-05-01','YYYY-MM-DD'))
tablespace ts_o40

partition ord41 values less than (to_date('1995-06-01','YYYY-MM-DD'))
tablespace ts_o41

partition ord42 values less than (to_date('1995-07-01','YYYY-MM-DD'))
tablespace ts_o42

partition ord43 values less than (to_date('1995-08-01','YYYY-MM-DD'))
tablespace ts_o43

partition ord44 values less than (to_date('1995-09-01','YYYY-MM-DD'))
tablespace ts_o44

partition ord45 values less than (to_date('1995-10-01','YYYY-MM-DD'))
tablespace ts_o45

partition ord46 values less than (to_date('1995-11-01','YYYY-MM-DD'))
tablespace ts_o46

partition ord47 values less than (to_date('1995-12-01','YYYY-MM-DD'))
tablespace ts_o47

partition ord48 values less than (to_date('1996-01-01','YYYY-MM-DD'))
tablespace ts_o48

partition ord49 values less than (to_date('1996-02-01','YYYY-MM-DD'))
tablespace ts_o49

partition ord50 values less than (to_date('1996-03-01','YYYY-MM-DD'))
```

```

tablespace ts_o50

partition ord51 values less than (to_date('1996-04-01','YYYY-MM-DD'))
tablespace ts_o51

partition ord52 values less than (to_date('1996-05-01','YYYY-MM-DD'))
tablespace ts_o52

partition ord53 values less than (to_date('1996-06-01','YYYY-MM-DD'))
tablespace ts_o53

partition ord54 values less than (to_date('1996-07-01','YYYY-MM-DD'))
tablespace ts_o54

partition ord55 values less than (to_date('1996-08-01','YYYY-MM-DD'))
tablespace ts_o55

partition ord56 values less than (to_date('1996-09-01','YYYY-MM-DD'))
tablespace ts_o56

partition ord57 values less than (to_date('1996-10-01','YYYY-MM-DD'))
tablespace ts_o57

partition ord58 values less than (to_date('1996-11-01','YYYY-MM-DD'))
tablespace ts_o58

partition ord59 values less than (to_date('1996-12-01','YYYY-MM-DD'))
tablespace ts_o59

partition ord60 values less than (to_date('1997-01-01','YYYY-MM-DD'))
tablespace ts_o60

partition ord61 values less than (to_date('1997-02-01','YYYY-MM-DD'))
tablespace ts_o61

partition ord62 values less than (to_date('1997-03-01','YYYY-MM-DD'))
tablespace ts_o62

partition ord63 values less than (to_date('1997-04-01','YYYY-MM-DD'))
tablespace ts_o63

partition ord64 values less than (to_date('1997-05-01','YYYY-MM-DD'))
tablespace ts_o64

partition ord65 values less than (to_date('1997-06-01','YYYY-MM-DD'))
tablespace ts_o65

partition ord66 values less than (to_date('1997-07-01','YYYY-MM-DD'))
tablespace ts_o66

partition ord67 values less than (to_date('1997-08-01','YYYY-MM-DD'))
tablespace ts_o67

partition ord68 values less than (to_date('1997-09-01','YYYY-MM-DD'))
tablespace ts_o68

partition ord69 values less than (to_date('1997-10-01','YYYY-MM-DD'))
tablespace ts_o69

partition ord70 values less than (to_date('1997-11-01','YYYY-MM-DD'))
tablespace ts_o70

partition ord71 values less than (to_date('1997-12-01','YYYY-MM-DD'))
tablespace ts_o71

partition ord72 values less than (to_date('1998-01-01','YYYY-MM-DD'))
tablespace ts_o72

partition ord73 values less than (to_date('1998-02-01','YYYY-MM-DD'))
tablespace ts_o73

partition ord74 values less than (to_date('1998-03-01','YYYY-MM-DD'))
tablespace ts_o74

partition ord75 values less than (to_date('1998-04-01','YYYY-MM-DD'))
tablespace ts_o75

partition ord76 values less than (to_date('1998-05-01','YYYY-MM-DD'))
tablespace ts_o76

partition ord77 values less than (to_date('1998-06-01','YYYY-MM-DD'))
tablespace ts_o77

partition ord78 values less than (to_date('1998-07-01','YYYY-MM-DD'))
tablespace ts_o78

partition ord79 values less than (to_date('1998-08-01','YYYY-MM-DD'))
tablespace ts_o79

partition ord80 values less than (MAXVALUE)
tablespace ts_o80
)

drop table partsupp;
create table partsupp (
    ps_partkey      number NOT NULL,
    ps_supplekey    number NOT NULL,
    ps_supplycost   number NOT NULL,
    ps_availqty     number ,
    ps_comment      varchar(199)
)

pctfree 0
pctused 99
tablespace ts_ps
storage (initial 64k next 295m pctincrease 0)
parallel (degree 60 )

drop table parts;
create table parts (
    p_partkey      number NOT NULL,
    p_type         varchar(25) ,
    p_size         number ,
    p_brand        varchar(10) ,
    p_name         varchar(55) ,
    p_container    varchar(10) ,
    p_mfg          varchar(25) ,
    p_retailprice   number ,
    p_comment      varchar(23)
)

pctfree 0
pctused 99
tablespace ts_p1
storage (initial 64k next 57m pctincrease 0)
parallel (degree 60 )
partition by range (p_size)
(
    partition part1 values less than (2)
    tablespace ts_p1

    partition part2 values less than (3)
    tablespace ts_p2

    partition part3 values less than (4)
    tablespace ts_p3
)

```

,	partition part4 values less than (5)	tablespace ts_p4	,	tablespace ts_p26
,	partition part5 values less than (6)	tablespace ts_p5	,	partition part27 values less than (28)
,	partition part6 values less than (7)	tablespace ts_p6	,	tablespace ts_p27
,	partition part7 values less than (8)	tablespace ts_p7	,	partition part28 values less than (29)
,	partition part8 values less than (9)	tablespace ts_p8	,	tablespace ts_p28
,	partition part9 values less than (10)	tablespace ts_p9	,	partition part29 values less than (30)
,	partition part10 values less than (11)	tablespace ts_p10	,	tablespace ts_p29
,	partition part11 values less than (12)	tablespace ts_p11	,	partition part30 values less than (31)
,	partition part12 values less than (13)	tablespace ts_p12	,	tablespace ts_p30
,	partition part13 values less than (14)	tablespace ts_p13	,	partition part31 values less than (32)
,	partition part14 values less than (15)	tablespace ts_p14	,	tablespace ts_p31
,	partition part15 values less than (16)	tablespace ts_p15	,	partition part32 values less than (33)
,	partition part16 values less than (17)	tablespace ts_p16	,	tablespace ts_p32
,	partition part17 values less than (18)	tablespace ts_p17	,	partition part33 values less than (34)
,	partition part18 values less than (19)	tablespace ts_p18	,	tablespace ts_p33
,	partition part19 values less than (20)	tablespace ts_p19	,	partition part34 values less than (35)
,	partition part20 values less than (21)	tablespace ts_p20	,	tablespace ts_p34
,	partition part21 values less than (22)	tablespace ts_p21	,	partition part35 values less than (36)
,	partition part22 values less than (23)	tablespace ts_p22	,	tablespace ts_p35
,	partition part23 values less than (24)	tablespace ts_p23	,	partition part36 values less than (37)
,	partition part24 values less than (25)	tablespace ts_p24	,	tablespace ts_p36
,	partition part25 values less than (26)	tablespace ts_p25	,	partition part37 values less than (38)
,	partition part26 values less than (27)		,	tablespace ts_p37
			,	partition part38 values less than (39)
			,	tablespace ts_p38
			,	partition part39 values less than (40)
			,	tablespace ts_p39
			,	partition part40 values less than (41)
			,	tablespace ts_p40
			,	partition part41 values less than (42)
			,	tablespace ts_p41
			,	partition part42 values less than (43)
			,	tablespace ts_p42
			,	partition part43 values less than (44)
			,	tablespace ts_p43
			,	partition part44 values less than (45)
			,	tablespace ts_p44
			,	partition part45 values less than (46)
			,	tablespace ts_p45
			,	partition part46 values less than (47)
			,	tablespace ts_p46
			,	partition part47 values less than (48)
			,	tablespace ts_p47
			,	partition part48 values less than (49)
			,	tablespace ts_p48
			,	

```

partition part49 values less than (50)
tablespace ts_p49
;
partition part50 values less than (MAXVALUE)
tablespace ts_p50
)
;
drop table customer;
create table customer (
    c_custkey          number NOT NULL,
    c_mktsegment       varchar(10) ,
    c_nationkey        number ,
    c_name             varchar(25) ,
    c_address          varchar(40) ,
    c_phone            varchar(15) ,
    c_acctbal          number ,
    c_comment          varchar(117)
)
pctfree 0
pctused 99
tablespace ts_c1
storage (initial 64k next 14m pctincrease 0)
parallel (degree 60 )
partition by range (c_mktsegment)
(
    partition cust1 values less than ('AUTOMOBILF')
    tablespace ts_c1
;
    partition cust2 values less than ('BUILDINH')
    tablespace ts_c2
;
    partition cust3 values less than ('FURNITURF')
    tablespace ts_c3
;
    partition cust4 values less than ('HOUSEHOLE')
    tablespace ts_c4
;
    partition cust5 values less than (MAXVALUE)
    tablespace ts_c5
)
;
drop table supplier;
create table supplier (
    s_suppkey          number NOT NULL,
    s_nationkey        number ,
    s_comment          varchar(101) ,
    s_name             varchar(25) ,
    s_address          varchar(40) ,
    s_phone            varchar(15) ,
    s_acctbal          number
)
pctfree 0
pctused 99
tablespace ts_s
storage (initial 64k next 30m pctincrease 0)
parallel (degree 60 )
;
drop table nation;
create table nation (
    n_nationkey        number NOT NULL,
    n_name             varchar(25) ,
    n_regionkey        number ,
    n_comment          varchar(152)
)
cache
tablespace system
;

```

```

data=z:/tpcd/data/hundgig/parts1.30 direct=true parallel=true
FILE=\\\\\\ORACLE\\DATA_PARS_30 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hundgig/parts31.ctl
log=c:/tpcd/log/hundgig/parts31.log
data=z:/tpcd/data/hundgig/parts1.31 direct=true parallel=true
FILE=\\\\\\ORACLE\\DATA_PARS_31 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hundgig/parts32.ctl
log=c:/tpcd/log/hundgig/parts32.log
data=z:/tpcd/data/hundgig/parts1.32 direct=true parallel=true
FILE=\\\\\\ORACLE\\DATA_PARS_32 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hundgig/parts33.ctl
log=c:/tpcd/log/hundgig/parts33.log
data=z:/tpcd/data/hundgig/parts1.33 direct=true parallel=true
FILE=\\\\\\ORACLE\\DATA_PARS_33 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hundgig/parts34.ctl
log=c:/tpcd/log/hundgig/parts34.log
data=z:/tpcd/data/hundgig/parts1.34 direct=true parallel=true
FILE=\\\\\\ORACLE\\DATA_PARS_34 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hundgig/parts35.ctl
log=c:/tpcd/log/hundgig/parts35.log
data=z:/tpcd/data/hundgig/parts1.35 direct=true parallel=true
FILE=\\\\\\ORACLE\\DATA_PARS_35 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hundgig/parts36.ctl
log=c:/tpcd/log/hundgig/parts36.log
data=z:/tpcd/data/hundgig/parts1.36 direct=true parallel=true
FILE=\\\\\\ORACLE\\DATA_PARS_36 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hundgig/parts37.ctl
log=c:/tpcd/log/hundgig/parts37.log
data=z:/tpcd/data/hundgig/parts1.37 direct=true parallel=true
FILE=\\\\\\ORACLE\\DATA_PARS_37 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hundgig/parts38.ctl
log=c:/tpcd/log/hundgig/parts38.log
data=z:/tpcd/data/hundgig/parts1.38 direct=true parallel=true
FILE=\\\\\\ORACLE\\DATA_PARS_38 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hundgig/parts39.ctl
log=c:/tpcd/log/hundgig/parts39.log
data=z:/tpcd/data/hundgig/parts1.39 direct=true parallel=true
FILE=\\\\\\ORACLE\\DATA_PARS_39 &
wait
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hundgig/parts40.ctl
log=c:/tpcd/log/hundgig/parts40.log
data=z:/tpcd/data/hundgig/parts1.40 direct=true parallel=true
FILE=\\\\\\ORACLE\\DATA_PARS_40 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hundgig/parts41.ctl
log=c:/tpcd/log/hundgig/parts41.log
data=z:/tpcd/data/hundgig/parts1.41 direct=true parallel=true
FILE=\\\\\\ORACLE\\DATA_PARS_41 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hundgig/parts42.ctl
log=c:/tpcd/log/hundgig/parts42.log
data=z:/tpcd/data/hundgig/parts1.42 direct=true parallel=true
FILE=\\\\\\ORACLE\\DATA_PARS_42 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hundgig/parts43.ctl
log=c:/tpcd/log/hundgig/parts43.log
data=z:/tpcd/data/hundgig/parts1.43 direct=true parallel=true
FILE=\\\\\\ORACLE\\DATA_PARS_43 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hundgig/parts44.ctl
log=c:/tpcd/log/hundgig/parts44.log
data=z:/tpcd/data/hundgig/parts1.44 direct=true parallel=true
FILE=\\\\\\ORACLE\\DATA_PARS_44 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hundgig/parts45.ctl
log=c:/tpcd/log/hundgig/parts45.log
data=z:/tpcd/data/hundgig/parts1.45 direct=true parallel=true
FILE=\\\\\\ORACLE\\DATA_PARS_45 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hundgig/parts46.ctl
log=c:/tpcd/log/hundgig/parts46.log
data=z:/tpcd/data/hundgig/parts1.46 direct=true parallel=true
FILE=\\\\\\ORACLE\\DATA_PARS_46 &

```


[illegible]

[illegible]


```
data=z:/tpcd/data/hungdig/customer28.5 direct=true parallel=true
FILE=//////////IIIORA_CUSTOMER5_28 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hungdig/customer141.ctl
log=c:/tpcd/log/hungdig/customer141.log
data=z:/tpcd/data/hungdig/customer29.1 direct=true parallel=true
FILE=//////////IIIORA_CUSTOMER1_29 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hungdig/customer142.ctl
log=c:/tpcd/log/hungdig/customer142.log
data=z:/tpcd/data/hungdig/customer29.2 direct=true parallel=true
FILE=//////////IIIORA_CUSTOMER2_29 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hungdig/customer143.ctl
log=c:/tpcd/log/hungdig/customer143.log
data=z:/tpcd/data/hungdig/customer29.3 direct=true parallel=true
FILE=//////////IIIORA_CUSTOMER3_29 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hungdig/customer144.ctl
log=c:/tpcd/log/hungdig/customer144.log
data=z:/tpcd/data/hungdig/customer29.4 direct=true parallel=true
FILE=//////////IIIORA_CUSTOMER4_29 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hungdig/customer145.ctl
log=c:/tpcd/log/hungdig/customer145.log
data=z:/tpcd/data/hungdig/customer29.5 direct=true parallel=true
FILE=//////////IIIORA_CUSTOMER5_29 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hungdig/customer146.ctl
log=c:/tpcd/log/hungdig/customer146.log
data=z:/tpcd/data/hungdig/customer30.1 direct=true parallel=true
FILE=//////////IIIORA_CUSTOMER1_30 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hungdig/customer147.ctl
log=c:/tpcd/log/hungdig/customer147.log
data=z:/tpcd/data/hungdig/customer30.2 direct=true parallel=true
FILE=//////////IIIORA_CUSTOMER2_30 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hungdig/customer148.ctl
log=c:/tpcd/log/hungdig/customer148.log
data=z:/tpcd/data/hungdig/customer30.3 direct=true parallel=true
FILE=//////////IIIORA_CUSTOMER3_30 &
wait
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hungdig/customer149.ctl
log=c:/tpcd/log/hungdig/customer149.log
data=z:/tpcd/data/hungdig/customer30.4 direct=true parallel=true
FILE=//////////IIIORA_CUSTOMER4_30 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hungdig/customer150.ctl
log=c:/tpcd/log/hungdig/customer150.log
data=z:/tpcd/data/hungdig/customer30.5 direct=true parallel=true
FILE=//////////IIIORA_CUSTOMER5_30 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hungdig/customer151.ctl
log=c:/tpcd/log/hungdig/customer151.log
data=z:/tpcd/data/hungdig/customer31.1 direct=true parallel=true
FILE=//////////IIIORA_CUSTOMER1_31 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hungdig/customer152.ctl
log=c:/tpcd/log/hungdig/customer152.log
data=z:/tpcd/data/hungdig/customer31.2 direct=true parallel=true
FILE=//////////IIIORA_CUSTOMER2_31 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hungdig/customer153.ctl
log=c:/tpcd/log/hungdig/customer153.log
data=z:/tpcd/data/hungdig/customer31.3 direct=true parallel=true
FILE=//////////IIIORA_CUSTOMER3_31 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hungdig/customer154.ctl
log=c:/tpcd/log/hungdig/customer154.log
data=z:/tpcd/data/hungdig/customer31.4 direct=true parallel=true
FILE=//////////IIIORA_CUSTOMER4_31 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hungdig/customer155.ctl
log=c:/tpcd/log/hungdig/customer155.log
data=z:/tpcd/data/hungdig/customer31.5 direct=true parallel=true
FILE=//////////IIIORA_CUSTOMER5_31 &
sqlldr80 tpcd/tpcd control=c:/tpcd/loadctl/hungdig/customer156.ctl
log=c:/tpcd/log/hungdig/customer156.log
data=z:/tpcd/data/hungdig/customer32.1 direct=true parallel=true
FILE=//////////IIIORA_CUSTOMER1_32 &
```


[illegible]

[illegible]


```
drop index I_osdqeys;  
create index I_osdqeys
```

```
partition losdqeps58
partition losdqeps59
```

```

partition losdqeps60,
partition losdqeps61,
partition losdqeps62,
partition losdqeps63,
partition losdqeps64,
partition losdqeps65,
partition losdqeps66,
partition losdqeps67,
partition losdqeps68,
partition losdqeps69,
partition losdqeps70,
partition losdqeps71,
partition losdqeps72,
partition losdqeps73,
partition losdqeps74,
partition losdqeps75,
partition losdqeps76,
partition losdqeps77,
partition losdqeps78,
partition losdqeps79,
partition losdqeps80
)
parallel (degree 60 instances 1)
;

drop index l_rossc;
create index l_rossc
on lineitem
(l_receiptdate,l_orderkey,l_shipmode,l_shipdate,l_commitdate)
pctfree 2
initrans 10
tablespace ts_ind5
storage (initial 5m next 5m freelists 16 freelist groups 5 maxextents
unlimited pctincrease 0)
global partition by range (l_receiptdate)
(
partition lrossc1 values less than (to_date('1992-02-01','YYYY-MM-
DD')),
partition lrossc2 values less than (to_date('1992-03-01','YYYY-MM-
DD')),
partition lrossc3 values less than (to_date('1992-04-01','YYYY-MM-
DD')),
partition lrossc4 values less than (to_date('1992-05-01','YYYY-MM-
DD')),
partition lrossc5 values less than (to_date('1992-06-01','YYYY-MM-
DD')),
partition lrossc6 values less than (to_date('1992-07-01','YYYY-MM-
DD')),
partition lrossc7 values less than (to_date('1992-08-01','YYYY-MM-
DD')),
partition lrossc8 values less than (to_date('1992-09-01','YYYY-MM-
DD')),
partition lrossc9 values less than (to_date('1992-10-01','YYYY-MM-
DD')),
partition lrossc10 values less than (to_date('1992-11-01','YYYY-MM-
DD')),
partition lrossc11 values less than (to_date('1992-12-01','YYYY-MM-
DD')),
partition lrossc12 values less than (to_date('1993-01-01','YYYY-MM-
DD')),
partition lrossc13 values less than (to_date('1993-02-01','YYYY-MM-
DD')),
partition lrossc14 values less than (to_date('1993-03-01','YYYY-MM-
DD')),
partition lrossc15 values less than (to_date('1993-04-01','YYYY-MM-
DD')),

```

```

partition lrossc16 values less than (to_date('1993-05-01','YYYY-MM-
DD')),
partition lrossc17 values less than (to_date('1993-06-01','YYYY-MM-
DD')),
partition lrossc18 values less than (to_date('1993-07-01','YYYY-MM-
DD')),
partition lrossc19 values less than (to_date('1993-08-01','YYYY-MM-
DD')),
partition lrossc20 values less than (to_date('1993-09-01','YYYY-MM-
DD')),
partition lrossc21 values less than (to_date('1993-10-01','YYYY-MM-
DD')),
partition lrossc22 values less than (to_date('1993-11-01','YYYY-MM-
DD')),
partition lrossc23 values less than (to_date('1993-12-01','YYYY-MM-
DD')),
partition lrossc24 values less than (to_date('1994-01-01','YYYY-MM-
DD')),
partition lrossc25 values less than (to_date('1994-02-01','YYYY-MM-
DD')),
partition lrossc26 values less than (to_date('1994-03-01','YYYY-MM-
DD')),
partition lrossc27 values less than (to_date('1994-04-01','YYYY-MM-
DD')),
partition lrossc28 values less than (to_date('1994-05-01','YYYY-MM-
DD')),
partition lrossc29 values less than (to_date('1994-06-01','YYYY-MM-
DD')),
partition lrossc30 values less than (to_date('1994-07-01','YYYY-MM-
DD')),
partition lrossc31 values less than (to_date('1994-08-01','YYYY-MM-
DD')),
partition lrossc32 values less than (to_date('1994-09-01','YYYY-MM-
DD')),
partition lrossc33 values less than (to_date('1994-10-01','YYYY-MM-
DD')),
partition lrossc34 values less than (to_date('1994-11-01','YYYY-MM-
DD')),
partition lrossc35 values less than (to_date('1994-12-01','YYYY-MM-
DD')),
partition lrossc36 values less than (to_date('1995-01-01','YYYY-MM-
DD')),
partition lrossc37 values less than (to_date('1995-02-01','YYYY-MM-
DD')),
partition lrossc38 values less than (to_date('1995-03-01','YYYY-MM-
DD')),
partition lrossc39 values less than (to_date('1995-04-01','YYYY-MM-
DD')),
partition lrossc40 values less than (to_date('1995-05-01','YYYY-MM-
DD')),
partition lrossc41 values less than (to_date('1995-06-01','YYYY-MM-
DD')),
partition lrossc42 values less than (to_date('1995-07-01','YYYY-MM-
DD')),
partition lrossc43 values less than (to_date('1995-08-01','YYYY-MM-
DD')),
partition lrossc44 values less than (to_date('1995-09-01','YYYY-MM-
DD')),
partition lrossc45 values less than (to_date('1995-10-01','YYYY-MM-
DD')),
partition lrossc46 values less than (to_date('1995-11-01','YYYY-MM-
DD')),
partition lrossc47 values less than (to_date('1995-12-01','YYYY-MM-
DD')),
partition lrossc48 values less than (to_date('1996-01-01','YYYY-MM-
DD')),

```

```

partition Irosc49 values less than (to_date('1996-02-01','YYYY-MM-DD')),
partition Irosc50 values less than (to_date('1996-03-01','YYYY-MM-DD')),
partition Irosc51 values less than (to_date('1996-04-01','YYYY-MM-DD')),
partition Irosc52 values less than (to_date('1996-05-01','YYYY-MM-DD')),
partition Irosc53 values less than (to_date('1996-06-01','YYYY-MM-DD')),
partition Irosc54 values less than (to_date('1996-07-01','YYYY-MM-DD')),
partition Irosc55 values less than (to_date('1996-08-01','YYYY-MM-DD')),
partition Irosc56 values less than (to_date('1996-09-01','YYYY-MM-DD')),
partition Irosc57 values less than (to_date('1996-10-01','YYYY-MM-DD')),
partition Irosc58 values less than (to_date('1996-11-01','YYYY-MM-DD')),
partition Irosc59 values less than (to_date('1996-12-01','YYYY-MM-DD')),
partition Irosc60 values less than (to_date('1997-01-01','YYYY-MM-DD')),
partition Irosc61 values less than (to_date('1997-02-01','YYYY-MM-DD')),
partition Irosc62 values less than (to_date('1997-03-01','YYYY-MM-DD')),
partition Irosc63 values less than (to_date('1997-04-01','YYYY-MM-DD')),
partition Irosc64 values less than (to_date('1997-05-01','YYYY-MM-DD')),
partition Irosc65 values less than (to_date('1997-06-01','YYYY-MM-DD')),
partition Irosc66 values less than (to_date('1997-07-01','YYYY-MM-DD')),
partition Irosc67 values less than (to_date('1997-08-01','YYYY-MM-DD')),
partition Irosc68 values less than (to_date('1997-09-01','YYYY-MM-DD')),
partition Irosc69 values less than (to_date('1997-10-01','YYYY-MM-DD')),
partition Irosc70 values less than (to_date('1997-11-01','YYYY-MM-DD')),
partition Irosc71 values less than (to_date('1997-12-01','YYYY-MM-DD')),
partition Irosc72 values less than (to_date('1998-01-01','YYYY-MM-DD')),
partition Irosc73 values less than (to_date('1998-02-01','YYYY-MM-DD')),
partition Irosc74 values less than (to_date('1998-03-01','YYYY-MM-DD')),
partition Irosc75 values less than (to_date('1998-04-01','YYYY-MM-DD')),
partition Irosc76 values less than (to_date('1998-05-01','YYYY-MM-DD')),
partition Irosc77 values less than (to_date('1998-06-01','YYYY-MM-DD')),
partition Irosc78 values less than (to_date('1998-07-01','YYYY-MM-DD')),
partition Irosc79 values less than (to_date('1998-08-01','YYYY-MM-DD')),
partition Irosc80 values less than (MAXVALUE))
parallel (degree 60 instances 1)
;

```

```

drop index I_ored;
create index I_ored

```

```

on lineitem (I_orderkey,I_returnflag,I_extendedprice,I_discount)
pctfree 2
initrans 10
tablespace ts_ind1
storage (initial 5m next 5m freelists 16 freelist groups 5 maxextents
unlimited pctincrease 0)
parallel (degree 60 instances 1)
;

```

```

drop index o_clokod;
create index o_clokod
on orders (o_clerk,o_orderkey,o_orderdate)
pctfree 2
initrans 10
tablespace ts_ind3
storage (initial 5m next 5m freelists 16 freelist groups 5 maxextents
unlimited pctincrease 0)
local (
partition oclokod1,
partition oclokod2,
partition oclokod3,
partition oclokod4,
partition oclokod5,
partition oclokod6,
partition oclokod7,
partition oclokod8,
partition oclokod9,
partition oclokod10,
partition oclokod11,
partition oclokod12,
partition oclokod13,
partition oclokod14,
partition oclokod15,
partition oclokod16,
partition oclokod17,
partition oclokod18,
partition oclokod19,
partition oclokod20,
partition oclokod21,
partition oclokod22,
partition oclokod23,
partition oclokod24,
partition oclokod25,
partition oclokod26,
partition oclokod27,
partition oclokod28,
partition oclokod29,
partition oclokod30,
partition oclokod31,
partition oclokod32,
partition oclokod33,
partition oclokod34,
partition oclokod35,
partition oclokod36,
partition oclokod37,
partition oclokod38,
partition oclokod39,
partition oclokod40,
partition oclokod41,
partition oclokod42,
partition oclokod43,
partition oclokod44,
partition oclokod45,
partition oclokod46,
partition oclokod47,
partition oclokod48,
partition oclokod49,
partition oclokod50,

```

```

partition oclokod51,
partition oclokod52,
partition oclokod53,
partition oclokod54,
partition oclokod55,
partition oclokod56,
partition oclokod57,
partition oclokod58,
partition oclokod59,
partition oclokod60,
partition oclokod61,
partition oclokod62,
partition oclokod63,
partition oclokod64,
partition oclokod65,
partition oclokod66,
partition oclokod67,
partition oclokod68,
partition oclokod69,
partition oclokod70,
partition oclokod71,
partition oclokod72,
partition oclokod73,
partition oclokod74,
partition oclokod75,
partition oclokod76,
partition oclokod77,
partition oclokod78,
partition oclokod79,
partition oclokod80
)
parallel (degree 60 instances 1)
;

drop index o_op;
create unique index o_op
on orders (o_orderkey,o_orderpriority)
pctfree 2
intrans 10
tablespace ts_ind3
storage (initial 5m next 5m freelists 16 freelist groups 5 maxextents
unlimited pctincrease 0)
parallel (degree 60 )
;

drop index ps_pksksc;
create unique index ps_pksksc
on partsupp (ps_partkey,ps_supkey,ps_supplycost)
pctfree 1
tablespace ts_ind4
storage (initial 1m next 1m maxextents unlimited pctincrease 0)
parallel (degree 60 )
;

drop index ps_spsa;
create unique index ps_spsa
on partsupp (ps_supkey,ps_partkey,ps_supplycost,ps_availqty)
pctfree 1
tablespace ts_ind4
storage (initial 1m next 1m maxextents unlimited pctincrease 0)
parallel (degree 60 )
;

drop index p_sptm;
create unique index p_sptm
on parts (p_size,p_partkey,p_type,p_mfgr)
pctfree 1

tablespace ts_ind5
storage (initial 1m next 1m maxextents unlimited pctincrease 0)
local (
partition psptm1,
partition psptm2,
partition psptm3,
partition psptm4,
partition psptm5,
partition psptm6,
partition psptm7,
partition psptm8,
partition psptm9,
partition psptm10,
partition psptm11,
partition psptm12,
partition psptm13,
partition psptm14,
partition psptm15,
partition psptm16,
partition psptm17,
partition psptm18,
partition psptm19,
partition psptm20,
partition psptm21,
partition psptm22,
partition psptm23,
partition psptm24,
partition psptm25,
partition psptm26,
partition psptm27,
partition psptm28,
partition psptm29,
partition psptm30,
partition psptm31,
partition psptm32,
partition psptm33,
partition psptm34,
partition psptm35,
partition psptm36,
partition psptm37,
partition psptm38,
partition psptm39,
partition psptm40,
partition psptm41,
partition psptm42,
partition psptm43,
partition psptm44,
partition psptm45,
partition psptm46,
partition psptm47,
partition psptm48,
partition psptm49,
partition psptm50
)
parallel (degree 60 instances 1)
;

drop index p_cbp;
create index p_cbp
on parts (p_container,p_brand,p_partkey)
pctfree 1
tablespace ts_ind4
storage (initial 1m next 1m maxextents unlimited pctincrease 0)
global partition by range (p_container,p_brand)
(
partition pcbp1 values less than ('JUMBO BAG','Brand#111'),
partition pcbp2 values less than ('JUMBO BAG','Brand#121'),

```

[illegible][illegible]

[illegible][illegible]

[illegible]

[illegible][illegible]

[illegible][illegible]

[illegible][illegible]

```

partition pcbp949 values less than ('WRAP JAR','Brand#541'),
partition pcbp950 values less than ('WRAP JAR','Brand#551'),
partition pcbp951 values less than ('WRAP PACK','Brand#111'),
partition pcbp952 values less than ('WRAP PACK','Brand#121'),
partition pcbp953 values less than ('WRAP PACK','Brand#131'),
partition pcbp954 values less than ('WRAP PACK','Brand#141'),
partition pcbp955 values less than ('WRAP PACK','Brand#151'),
partition pcbp956 values less than ('WRAP PACK','Brand#211'),
partition pcbp957 values less than ('WRAP PACK','Brand#221'),
partition pcbp958 values less than ('WRAP PACK','Brand#231'),
partition pcbp959 values less than ('WRAP PACK','Brand#241'),
partition pcbp960 values less than ('WRAP PACK','Brand#251'),
partition pcbp961 values less than ('WRAP PACK','Brand#311'),
partition pcbp962 values less than ('WRAP PACK','Brand#321'),
partition pcbp963 values less than ('WRAP PACK','Brand#331'),
partition pcbp964 values less than ('WRAP PACK','Brand#341'),
partition pcbp965 values less than ('WRAP PACK','Brand#351'),
partition pcbp966 values less than ('WRAP PACK','Brand#411'),
partition pcbp967 values less than ('WRAP PACK','Brand#421'),
partition pcbp968 values less than ('WRAP PACK','Brand#431'),
partition pcbp969 values less than ('WRAP PACK','Brand#441'),
partition pcbp970 values less than ('WRAP PACK','Brand#451'),
partition pcbp971 values less than ('WRAP PACK','Brand#511'),
partition pcbp972 values less than ('WRAP PACK','Brand#521'),
partition pcbp973 values less than ('WRAP PACK','Brand#531'),
partition pcbp974 values less than ('WRAP PACK','Brand#541'),
partition pcbp975 values less than ('WRAP PACK','Brand#551'),
partition pcbp976 values less than ('WRAP PKG','Brand#111'),
partition pcbp977 values less than ('WRAP PKG','Brand#121'),
partition pcbp978 values less than ('WRAP PKG','Brand#131'),
partition pcbp979 values less than ('WRAP PKG','Brand#141'),
partition pcbp980 values less than ('WRAP PKG','Brand#151'),
partition pcbp981 values less than ('WRAP PKG','Brand#211'),
partition pcbp982 values less than ('WRAP PKG','Brand#221'),
partition pcbp983 values less than ('WRAP PKG','Brand#231'),
partition pcbp984 values less than ('WRAP PKG','Brand#241'),
partition pcbp985 values less than ('WRAP PKG','Brand#251'),
partition pcbp986 values less than ('WRAP PKG','Brand#311'),
partition pcbp987 values less than ('WRAP PKG','Brand#321'),
partition pcbp988 values less than ('WRAP PKG','Brand#331'),
partition pcbp989 values less than ('WRAP PKG','Brand#341'),
partition pcbp990 values less than ('WRAP PKG','Brand#351'),
partition pcbp991 values less than ('WRAP PKG','Brand#411'),
partition pcbp992 values less than ('WRAP PKG','Brand#421'),
partition pcbp993 values less than ('WRAP PKG','Brand#431'),
partition pcbp994 values less than ('WRAP PKG','Brand#441'),
partition pcbp995 values less than ('WRAP PKG','Brand#451'),
partition pcbp996 values less than ('WRAP PKG','Brand#511'),
partition pcbp997 values less than ('WRAP PKG','Brand#521'),
partition pcbp998 values less than ('WRAP PKG','Brand#531'),
partition pcbp999 values less than ('WRAP PKG','Brand#541'),
partition pcbp1000 values less than (MAXVALUE,MAXVALUE))
parallel (degree 60 )
;

```

```

drop index p_tp;
create index p_tp
on parts (p_type,p_partkey)
pctfree 1
tablespace ts_ind5
storage (initial 1m next 1m maxextents unlimited pctincrease 0)
global partition by range (p_type)
(
partition plp1 values less than ('ECONOMY ANODIZED COPPER'),
partition plp2 values less than ('ECONOMY ANODIZED NICKEL'),
partition plp3 values less than ('ECONOMY ANODIZED STEEL'),
partition plp4 values less than ('ECONOMY ANODIZED TIN'),

```

```

partition plp5 values less than ('ECONOMY BRUSHED BRASS'),
partition plp6 values less than ('ECONOMY BRUSHED COPPER'),
partition plp7 values less than ('ECONOMY BRUSHED NICKEL'),
partition plp8 values less than ('ECONOMY BRUSHED STEEL'),
partition plp9 values less than ('ECONOMY BRUSHED TIN'),
partition plp10 values less than ('ECONOMY BURNISHED BRASS'),
partition plp11 values less than ('ECONOMY BURNISHED COPPER'),
partition plp12 values less than ('ECONOMY BURNISHED NICKEL'),
partition plp13 values less than ('ECONOMY PLATED STEEL'),
partition plp14 values less than ('ECONOMY BURNISHED TIN'),
partition plp15 values less than ('ECONOMY PLATED BRASS'),
partition plp16 values less than ('ECONOMY PLATED COPPER'),
partition plp17 values less than ('ECONOMY PLATED NICKEL'),
partition plp18 values less than ('ECONOMY PLATED STEEL'),
partition plp19 values less than ('ECONOMY PLATED TIN'),
partition plp20 values less than ('ECONOMY POLISHED BRASS'),
partition plp21 values less than ('ECONOMY POLISHED COPPER'),
partition plp22 values less than ('ECONOMY POLISHED NICKEL'),
partition plp23 values less than ('ECONOMY POLISHED STEEL'),
partition plp24 values less than ('ECONOMY POLISHED TIN'),
partition plp25 values less than ('LARGE ANODIZED BRASS'),
partition plp26 values less than ('LARGE ANODIZED COPPER'),
partition plp27 values less than ('LARGE ANODIZED NICKEL'),
partition plp28 values less than ('LARGE ANODIZED STEEL'),
partition plp29 values less than ('LARGE ANODIZED TIN'),
partition plp30 values less than ('LARGE BRUSHED BRASS'),
partition plp31 values less than ('LARGE BRUSHED COPPER'),
partition plp32 values less than ('LARGE BRUSHED NICKEL'),
partition plp33 values less than ('LARGE BRUSHED STEEL'),
partition plp34 values less than ('LARGE BRUSHED TIN'),
partition plp35 values less than ('LARGE BURNISHED BRASS'),
partition plp36 values less than ('LARGE BURNISHED COPPER'),
partition plp37 values less than ('LARGE BURNISHED NICKEL'),
partition plp38 values less than ('LARGE BURNISHED STEEL'),
partition plp39 values less than ('LARGE BURNISHED TIN'),
partition plp40 values less than ('LARGE PLATED BRASS'),
partition plp41 values less than ('LARGE PLATED COPPER'),
partition plp42 values less than ('LARGE PLATED NICKEL'),
partition plp43 values less than ('LARGE PLATED STEEL'),
partition plp44 values less than ('LARGE PLATED TIN'),
partition plp45 values less than ('LARGE POLISHED BRASS'),
partition plp46 values less than ('LARGE POLISHED COPPER'),
partition plp47 values less than ('LARGE POLISHED NICKEL'),
partition plp48 values less than ('LARGE POLISHED STEEL'),
partition plp49 values less than ('LARGE POLISHED TIN'),
partition plp50 values less than ('MEDIUM ANODIZED BRASS'),
partition plp51 values less than ('MEDIUM ANODIZED COPPER'),
partition plp52 values less than ('MEDIUM ANODIZED NICKEL'),
partition plp53 values less than ('MEDIUM ANODIZED STEEL'),
partition plp54 values less than ('MEDIUM ANODIZED TIN'),
partition plp55 values less than ('MEDIUM BRUSHED BRASS'),
partition plp56 values less than ('MEDIUM BRUSHED COPPER'),
partition plp57 values less than ('MEDIUM BRUSHED NICKEL'),
partition plp58 values less than ('MEDIUM BRUSHED STEEL'),
partition plp59 values less than ('MEDIUM BRUSHED TIN'),
partition plp60 values less than ('MEDIUM BURNISHED BRASS'),
partition plp61 values less than ('MEDIUM BURNISHED COPPER'),
partition plp62 values less than ('MEDIUM BURNISHED NICKEL'),
partition plp63 values less than ('MEDIUM BURNISHED STEEL'),
partition plp64 values less than ('MEDIUM BURNISHED TIN'),
partition plp65 values less than ('MEDIUM PLATED BRASS'),
partition plp66 values less than ('MEDIUM PLATED COPPER'),
partition plp67 values less than ('MEDIUM PLATED NICKEL'),
partition plp68 values less than ('MEDIUM PLATED STEEL'),
partition plp69 values less than ('MEDIUM PLATED TIN'),
partition plp70 values less than ('MEDIUM POLISHED BRASS'),
partition plp71 values less than ('MEDIUM POLISHED COPPER'),

```

```
partition ptp72 values less than ('MEDIUM POLISHED NICKEL'),
partition ptp73 values less than ('MEDIUM POLISHED STEEL'),
partition ptp74 values less than ('MEDIUM POLISHED TIN'),
partition ptp75 values less than ('PROMO ANODIZED BRASS'),
partition ptp76 values less than ('PROMO ANODIZED COPPER'),
partition ptp77 values less than ('PROMO ANODIZED NICKEL'),
partition ptp78 values less than ('PROMO ANODIZED STEEL'),
partition ptp79 values less than ('PROMO ANODIZED TIN'),
partition ptp80 values less than ('PROMO BRUSHED BRASS'),
partition ptp81 values less than ('PROMO BRUSHED COPPER'),
partition ptp82 values less than ('PROMO BRUSHED NICKEL'),
partition ptp83 values less than ('PROMO BRUSHED STEEL'),
partition ptp84 values less than ('PROMO BRUSHED TIN'),
partition ptp85 values less than ('PROMO BURNISHED BRASS'),
partition ptp86 values less than ('PROMO BURNISHED COPPER'),
partition ptp87 values less than ('PROMO BURNISHED NICKEL'),
partition ptp88 values less than ('PROMO BURNISHED STEEL'),
partition ptp89 values less than ('PROMO BURNISHED TIN'),
partition ptp90 values less than ('PROMO PLATED BRASS'),
partition ptp91 values less than ('PROMO PLATED COPPER'),
partition ptp92 values less than ('PROMO PLATED NICKEL'),
partition ptp93 values less than ('PROMO PLATED STEEL'),
partition ptp94 values less than ('PROMO PLATED TIN'),
partition ptp95 values less than ('PROMO POLISHED BRASS'),
partition ptp96 values less than ('PROMO POLISHED COPPER'),
partition ptp97 values less than ('PROMO POLISHED NICKEL'),
partition ptp98 values less than ('PROMO POLISHED STEEL'),
partition ptp99 values less than ('PROMO POLISHED TIN'),
partition ptp100 values less than ('SMALL ANODIZED BRASS'),
partition ptp101 values less than ('SMALL ANODIZED COPPER'),
partition ptp102 values less than ('SMALL ANODIZED NICKEL'),
partition ptp103 values less than ('SMALL ANODIZED STEEL'),
partition ptp104 values less than ('SMALL ANODIZED TIN'),
partition ptp105 values less than ('SMALL BRUSHED BRASS'),
partition ptp106 values less than ('SMALL BRUSHED COPPER'),
partition ptp107 values less than ('SMALL BRUSHED NICKEL'),
partition ptp108 values less than ('SMALL BRUSHED STEEL'),
partition ptp109 values less than ('SMALL BRUSHED TIN'),
partition ptp110 values less than ('SMALL BURNISHED BRASS'),
partition ptp111 values less than ('SMALL BURNISHED COPPER'),
partition ptp112 values less than ('SMALL BURNISHED NICKEL'),
partition ptp113 values less than ('SMALL BURNISHED STEEL'),
partition ptp114 values less than ('SMALL BURNISHED TIN'),
partition ptp115 values less than ('SMALL PLATED BRASS'),
partition ptp116 values less than ('SMALL PLATED COPPER'),
partition ptp117 values less than ('SMALL PLATED NICKEL'),
partition ptp118 values less than ('SMALL PLATED STEEL'),
partition ptp119 values less than ('SMALL PLATED TIN'),
partition ptp120 values less than ('SMALL POLISHED BRASS'),
partition ptp121 values less than ('SMALL POLISHED COPPER'),
partition ptp122 values less than ('SMALL POLISHED NICKEL'),
partition ptp123 values less than ('SMALL POLISHED STEEL'),
partition ptp124 values less than ('SMALL POLISHED TIN'),
partition ptp125 values less than ('STANDARD ANODIZED BRASS'),
partition ptp126 values less than ('STANDARD ANODIZED COPPER'),
partition ptp127 values less than ('STANDARD ANODIZED NICKEL'),
partition ptp128 values less than ('STANDARD ANODIZED STEEL'),
partition ptp129 values less than ('STANDARD ANODIZED TIN'),
partition ptp130 values less than ('STANDARD BRUSHED BRASS'),
partition ptp131 values less than ('STANDARD BRUSHED COPPER'),
partition ptp132 values less than ('STANDARD BRUSHED NICKEL'),
partition ptp133 values less than ('STANDARD BRUSHED STEEL'),
partition ptp134 values less than ('STANDARD BRUSHED TIN'),
partition ptp135 values less than ('STANDARD BURNISHED BRASS'),
partition ptp136 values less than ('STANDARD BURNISHED
COPPER'),
partition ptp137 values less than ('STANDARD BURNISHED
NICKEL'),
```

```
partition ptp138 values less than ('STANDARD BURNISHED STEEL'),
partition ptp139 values less than ('STANDARD BURNISHED TIN'),
partition ptp140 values less than ('STANDARD PLATED BRASS'),
partition ptp141 values less than ('STANDARD PLATED COPPER'),
partition ptp142 values less than ('STANDARD PLATED NICKEL'),
partition ptp143 values less than ('STANDARD PLATED STEEL'),
partition ptp144 values less than ('STANDARD PLATED TIN'),
partition ptp145 values less than ('STANDARD POLISHED BRASS'),
partition ptp146 values less than ('STANDARD POLISHED COPPER'),
partition ptp147 values less than ('STANDARD POLISHED NICKEL'),
partition ptp148 values less than ('STANDARD POLISHED STEEL'),
partition ptp149 values less than ('STANDARD POLISHED TIN'),
partition ptp150 values less than (MAXVALUE))
parallel (degree 60 )
;
```

```
drop index s_skey;
create unique index s_skey
on supplier (s_suppkey)
pctfree 2
tablespace ts_ind4
storage (initial 1m next 1m maxextents unlimited pctincrease 0)
parallel (degree 60 )
;
```

```
drop index s_sn;
create unique index s_sn
on supplier (s_suppkey,s_nationkey)
pctfree 2
tablespace ts_ind4
storage (initial 1m next 1m maxextents unlimited pctincrease 0)
parallel (degree 60 instances 1)
;
```

```
drop index ps_pksk;
create unique index ps_pksk
on partsupp (ps_partkey, ps_suppkey)
pctfree 2
tablespace ts_ind5
storage (initial 7M next 7M pctincrease 0 maxextents unlimited)
parallel (degree 60 instances 1)
;
```

```
spool off
!
```

```
#####
#####
# Analyze Phase
echo "begin analyzing at 'date'"
```

```
plus80 tpcd/tpcd @anlyz index l_ored &
plus80 tpcd/tpcd @anlyzp index o_clokod oclokod1 &
plus80 tpcd/tpcd @anlyzp index o_clokod oclokod2 &
plus80 tpcd/tpcd @anlyzp index o_clokod oclokod3 &
plus80 tpcd/tpcd @anlyzp index o_clokod oclokod4 &
plus80 tpcd/tpcd @anlyzp index o_clokod oclokod5 &
plus80 tpcd/tpcd @anlyzp index o_clokod oclokod6 &
plus80 tpcd/tpcd @anlyzp index o_clokod oclokod7 &
plus80 tpcd/tpcd @anlyzp index o_clokod oclokod8 &
plus80 tpcd/tpcd @anlyzp index o_clokod oclokod9 &
plus80 tpcd/tpcd @anlyzp index o_clokod oclokod10 &
plus80 tpcd/tpcd @anlyzp index o_clokod oclokod11 &
plus80 tpcd/tpcd @anlyzp index o_clokod oclokod12 &
plus80 tpcd/tpcd @anlyzp index o_clokod oclokod13 &
plus80 tpcd/tpcd @anlyzp index o_clokod oclokod14 &
plus80 tpcd/tpcd @anlyzp index o_clokod oclokod15 &
plus80 tpcd/tpcd @anlyzp index o_clokod oclokod16 &
```


[illegible]

[illegible][illegible]

[illegible][illegible]

[illegible]


```
plus80 tpcd/tpcd @anlyzp table orders ord1 &
plus80 tpcd/tpcd @anlyzp table orders ord2 &
```

```
plus80 tpcd/tpcd @anlyzp table orders ord68 &
plus80 tpcd/tpcd @anlyzp table orders ord69 &
```

```

plus80 tpcd/tpcd @anlyzp table orders ord70 &
plus80 tpcd/tpcd @anlyzp table orders ord71 &
plus80 tpcd/tpcd @anlyzp table orders ord72 &
plus80 tpcd/tpcd @anlyzp table orders ord73 &
plus80 tpcd/tpcd @anlyzp table orders ord74 &
plus80 tpcd/tpcd @anlyzp table orders ord75 &
plus80 tpcd/tpcd @anlyzp table orders ord76 &
plus80 tpcd/tpcd @anlyzp table orders ord77 &
plus80 tpcd/tpcd @anlyzp table orders ord78 &
plus80 tpcd/tpcd @anlyzp table orders ord79 &
plus80 tpcd/tpcd @anlyzp table orders ord80 &
wait
plus80 tpcd/tpcd @anlyzp table partsupp &
plus80 tpcd/tpcd @anlyzp table parts part1 &
plus80 tpcd/tpcd @anlyzp table parts part2 &
plus80 tpcd/tpcd @anlyzp table parts part3 &
plus80 tpcd/tpcd @anlyzp table parts part4 &
plus80 tpcd/tpcd @anlyzp table parts part5 &
plus80 tpcd/tpcd @anlyzp table parts part6 &
plus80 tpcd/tpcd @anlyzp table parts part7 &
plus80 tpcd/tpcd @anlyzp table parts part8 &
plus80 tpcd/tpcd @anlyzp table parts part9 &
plus80 tpcd/tpcd @anlyzp table parts part10 &
plus80 tpcd/tpcd @anlyzp table parts part11 &
plus80 tpcd/tpcd @anlyzp table parts part12 &
plus80 tpcd/tpcd @anlyzp table parts part13 &
plus80 tpcd/tpcd @anlyzp table parts part14 &
plus80 tpcd/tpcd @anlyzp table parts part15 &
plus80 tpcd/tpcd @anlyzp table parts part16 &
plus80 tpcd/tpcd @anlyzp table parts part17 &
plus80 tpcd/tpcd @anlyzp table parts part18 &
plus80 tpcd/tpcd @anlyzp table parts part19 &
plus80 tpcd/tpcd @anlyzp table parts part20 &
plus80 tpcd/tpcd @anlyzp table parts part21 &
plus80 tpcd/tpcd @anlyzp table parts part22 &
plus80 tpcd/tpcd @anlyzp table parts part23 &
plus80 tpcd/tpcd @anlyzp table parts part24 &
plus80 tpcd/tpcd @anlyzp table parts part25 &
wait
plus80 tpcd/tpcd @anlyzp table parts part26 &
plus80 tpcd/tpcd @anlyzp table parts part27 &
plus80 tpcd/tpcd @anlyzp table parts part28 &
plus80 tpcd/tpcd @anlyzp table parts part29 &
plus80 tpcd/tpcd @anlyzp table parts part30 &
plus80 tpcd/tpcd @anlyzp table parts part31 &
plus80 tpcd/tpcd @anlyzp table parts part32 &
plus80 tpcd/tpcd @anlyzp table parts part33 &
plus80 tpcd/tpcd @anlyzp table parts part34 &
plus80 tpcd/tpcd @anlyzp table parts part35 &
plus80 tpcd/tpcd @anlyzp table parts part36 &
plus80 tpcd/tpcd @anlyzp table parts part37 &
plus80 tpcd/tpcd @anlyzp table parts part38 &
plus80 tpcd/tpcd @anlyzp table parts part39 &
plus80 tpcd/tpcd @anlyzp table parts part40 &
plus80 tpcd/tpcd @anlyzp table parts part41 &
plus80 tpcd/tpcd @anlyzp table parts part42 &
plus80 tpcd/tpcd @anlyzp table parts part43 &
plus80 tpcd/tpcd @anlyzp table parts part44 &
plus80 tpcd/tpcd @anlyzp table parts part45 &
plus80 tpcd/tpcd @anlyzp table parts part46 &
plus80 tpcd/tpcd @anlyzp table parts part47 &
plus80 tpcd/tpcd @anlyzp table parts part48 &
plus80 tpcd/tpcd @anlyzp table parts part49 &
plus80 tpcd/tpcd @anlyzp table parts part50 &
wait
plus80 tpcd/tpcd @anlyzp table customer cust1 &
plus80 tpcd/tpcd @anlyzp table customer cust2 &

```

```

plus80 tpcd/tpcd @anlyzp table customer cust3 &
plus80 tpcd/tpcd @anlyzp table customer cust4 &
plus80 tpcd/tpcd @anlyzp table customer cust5 &
plus80 tpcd/tpcd @anlyzp table supplier &
plus80 tpcd/tpcd @anlyzp table nation &
plus80 tpcd/tpcd @anlyzp table region &
wait
echo "done analyzing at `date`"

```

```

##### Restart database
svrmgr30 <<|
connect internal/manager
shutdown normal;
!

```

```

net stop oracleservice(tpcd)
net start oracleservice(tpcd)

```

```

svrmgr30 <<|
connect internal/manager;
startup pfile=/tpcd/dbs/hundgig/susan.ora;
alter tablespace ts_temp default storage (initial 5m next 5m
pctincrease 0);
connect tpcd/tpcd
alter table orders storage (next 5M pctincrease 0);
alter table lineitem storage (next 5M pctincrease 0);

```

```

connect tpcd/tpcd
set termout on;
set echo on;
spool \tpcd\hundgig\alter_para_803.log
alter table customer parallel (degree 60 instances 1);
alter table lineitem parallel (degree 60 instances 1);
alter table orders parallel (degree 60 instances 1);
alter table partsupp parallel (degree 1 instances 1);
alter table parts parallel (degree 60 instances 1);
alter table supplier parallel (degree 60 instances 1);
alter table nation parallel (degree 1 instances 1);
alter table region parallel (degree 1 instances 1);

```

```

alter index l_ored parallel (degree 60 instances 1);
alter index l_pqesod parallel (degree 1 instances 1);
alter index l_rossc parallel (degree 60 instances 1);
alter index o_clokod parallel (degree 60 instances 1);
alter index o_op parallel (degree 60 instances 1);
alter index ps_pksk parallel (degree 20 instances 1);
alter index ps_pksksc parallel (degree 20 instances 1);
alter index ps_spsa parallel (degree 1 instances 1);
alter index p_sptm parallel (degree 60 instances 1);
alter index p_tp parallel (degree 60 instances 1);
alter index p_cbp parallel (degree 60 instances 1);
alter index s_skey parallel (degree 60 instances 1);
alter index s_sn parallel (degree 60 instances 1);

```

```

select table_name, degree from user_tables;
select index_name, degree from user_indexes;

```

```

select to_char(sysdate, 'MM/DD/YYYY HH24:MI:SS') from dual;
spool off
!
echo "Done loading database and ready to run at `date`"
gtime
echo "Get time stamp, shutdown database, shutdown server,"
echo "remove flatfiles, restart server bring up database, "
echo "and get final timestamp and start 3 power runs."

```

ADDFILE.SQL

alter tablespace &1 add datafile '&2' size &3 reuse;

ANLYZ.SQL

alter tablespace &1 add datafile '&2' size &3 reuse;

ANLYZP.SQL

analyze &1 &2 partition (&3) estimate statistics sample 10 percent;

Appendix C: Query Text and Query Output

Validation Query EQT

```
-- TPC-D Parameter Substitution (Version 1.2.0 )
-- using default substitutions
-- Query 1 Original
SELECT
  L_RETURNFLAG,
  L_LINestatus,
  SUM(L_QUANTITY) AS SUM_QTY,
  SUM(L_EXTENDEDPRICE) AS SUM_BASE_PRICE,
  SUM(L_EXTENDEDPRICE * (1 - L_DISCOUNT)) AS
  SUM_DISC_PRICE,
  SUM(L_EXTENDEDPRICE * (1 - L_DISCOUNT) * (1 + L_TAX)) AS
  SUM_CHARGE,
  AVG(L_QUANTITY) AS AVG_QTY,
  AVG(L_EXTENDEDPRICE) AS AVG_PRICE,
  AVG(L_DISCOUNT) AS AVG_DISC,
  COUNT(*) AS COUNT_ORDER
FROM LINEITEM
WHERE
  L_SHIPDATE <= TO_DATE('1998-12-01','YYYY-MM-DD') - 90
GROUP BY L_RETURNFLAG, L_LINestatus
ORDER BY L_RETURNFLAG, L_LINestatus;
-- Query 4 (Original)
SELECT
  O_ORDERPRIORITY,
  COUNT(*) AS ORDER_COUNT
FROM ORDERS
WHERE
  O_ORDERDATE >= TO_DATE('1993-07-01','YYYY-MM-DD')
  AND O_ORDERDATE < ADD_MONTHS(TO_DATE('1993-07-
01','YYYY-MM-DD'),3)
  AND EXISTS
    (SELECT
      *
      FROM LINEITEM
      WHERE
        L_ORDERKEY = O_ORDERKEY
        AND L_COMMITDATE < L_RECEIPTDATE
    )
GROUP BY O_ORDERPRIORITY
ORDER BY O_ORDERPRIORITY;
-- Query 15 (Variant B)
CREATE TABLE REVENUE0 (SUPPLIER_NO INTEGER,
TOTAL_REVENUE NUMBER(20, 2))
tablespace TS_S
;

INSERT INTO REVENUE0
SELECT
  L_SUPPKEY,
  SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT))
FROM LINEITEM
WHERE
  L_SHIPDATE >= TO_DATE('1996-01-01','YYYY-MM-DD')
  AND L_SHIPDATE < ADD_MONTHS(TO_DATE('1996-01-01','YYYY-
MM-DD'),3)
GROUP BY L_SUPPKEY;
```

```
SELECT
  S_SUPPKEY,
  S_NAME,
  S_ADDRESS,
  S_PHONE,
  TOTAL_REVENUE
FROM SUPPLIER, REVENUE0
WHERE
  S_SUPPKEY = SUPPLIER_NO
  AND TOTAL_REVENUE =
    (SELECT
      MAX(TOTAL_REVENUE)
      FROM REVENUE0
    )
ORDER BY S_SUPPKEY;
```

```
DROP TABLE REVENUE0;
```

```
-- Query 10
set_fetchrows=20;
SELECT
  C_CUSTKEY,
  C_NAME,
  SUM(L_EXTENDEDPRICE * (1-L_DISCOUNT)) AS REVENUE,
  C_ACCTBAL,
  N_NAME,
  C_ADDRESS,
  C_PHONE,
  C_COMMENT
FROM CUSTOMER, ORDERS, LINEITEM, NATION
WHERE
  C_CUSTKEY = O_CUSTKEY
  AND L_ORDERKEY = O_ORDERKEY
  AND O_ORDERDATE >= TO_DATE('1993-10-01','YYYY-MM-DD')
  AND O_ORDERDATE < ADD_MONTHS(TO_DATE('1993-10-01'
,'YYYY-MM-DD'), 3)
  AND L_RETURNFLAG = 'R'
  AND C_NATIONKEY = N_NATIONKEY
GROUP BY C_CUSTKEY, C_NAME, C_ACCTBAL, C_PHONE,
N_NAME, C_ADDRESS, C_COMMENT
ORDER BY REVENUE DESC;
-- Query 11 (Variant A)
CREATE TABLE PART_VALUE0 (PARTNO INTEGER, VALUE
NUMBER(20, 2) )
tablespace TS_S
;
```

```
INSERT INTO PART_VALUE0
SELECT
  PS_PARTKEY,
  SUM(PS_SUPPLYCOST*PS_AVAILQTY)
FROM PARTSUPP, SUPPLIER, NATION
WHERE
  PS_SUPPKEY = S_SUPPKEY
  AND S_NATIONKEY = N_NATIONKEY
  AND N_NAME = 'GERMANY'
GROUP BY PS_PARTKEY;
```

```
CREATE TABLE SUM_PART_VALUE0 (TOTAL_VALUE
NUMBER(20, 2))
tablespace TS_S
;
```

```
INSERT INTO SUM_PART_VALUE0
SELECT
  SUM(PS_SUPPLYCOST*PS_AVAILQTY) * 0.0010000000
```

```

FROM PARTSUPP, SUPPLIER, NATION
WHERE
  PS_SUPPKEY = S_SUPPKEY
  AND S_NATIONKEY = N_NATIONKEY
  AND N_NAME = 'GERMANY';

SELECT
  PARTNO AS P_PARTKEY,
  VALUE
FROM PART_VALUE0
WHERE
  VALUE >
  (SELECT
    SUM(TOTAL_VALUE)
  FROM SUM_PART_VALUE0
  )
ORDER BY VALUE DESC;

DROP TABLE PART_VALUE0;

DROP TABLE SUM_PART_VALUE0;
-- Query 6
SELECT
  SUM(L_EXTENDEDPRI * L_DISCOUNT) AS REVENUE
FROM LINEITEM
WHERE
  L_SHIPDATE >= TO_DATE('1994-01-01','YYYY-MM-DD')
  AND L_SHIPDATE < ADD_MONTHS(TO_DATE('1994-01-01','YYYY-
MM-DD'),12)
  AND L_DISCOUNT BETWEEN .06 - 0.01 AND .06 + 0.01
  AND L_QUANTITY < 24;
-- Query 2 Variant A
set_fetchrows=100;
SELECT
  S_ACCTBAL,
  S_NAME,
  N_NAME,
  P_PARTKEY,
  P_MFGR,
  S_ADDRESS,
  S_PHONE,
  S_COMMENT
FROM PARTS, SUPPLIER, PARTSUPP, NATION, REGION
WHERE
  P_PARTKEY = PS_PARTKEY
  AND S_SUPPKEY = PS_SUPPKEY
  AND P_SIZE = 15
  AND P_TYPE LIKE '%BRASS'
  AND S_NATIONKEY = N_NATIONKEY
  AND N_REGIONKEY = R_REGIONKEY
  AND R_NAME = 'EUROPE'
  AND (P_PARTKEY,PS_SUPPLYCOST) IN
  (SELECT
    PS_PARTKEY,
    MIN(PS_SUPPLYCOST)
  FROM PARTSUPP, SUPPLIER, NATION, REGION
  WHERE
    S_SUPPKEY = PS_SUPPKEY
    AND S_NATIONKEY = N_NATIONKEY
    AND N_REGIONKEY = R_REGIONKEY
    AND R_NAME = 'EUROPE'
  GROUP BY PS_PARTKEY
  )
ORDER BY S_ACCTBAL DESC, N_NAME, S_NAME, P_PARTKEY;
-- Query 16
SELECT
  P_BRAND,
  P_TYPE,
  P_SIZE,
  COUNT(DISTINCT PS_SUPPKEY) AS SUPPLIER_CNT
FROM PARTSUPP,PARTS
WHERE
  P_PARTKEY = PS_PARTKEY
  AND P_BRAND <> 'Brand#45'
  AND P_TYPE NOT LIKE 'MEDIUM POLISHED%'
  AND P_SIZE IN (49, 14, 23, 45, 19, 3, 36, 9)
  AND PS_SUPPKEY NOT IN
  (SELECT
    S_SUPPKEY
  FROM SUPPLIER
  WHERE
    S_COMMENT LIKE '%Better Business Bureau%Complaints%'
  )
GROUP BY P_BRAND, P_TYPE, P_SIZE
ORDER BY SUPPLIER_CNT DESC, P_BRAND, P_TYPE, P_SIZE;
-- Query 14 (Variant C)
CREATE TABLE ALL_SALES0 (TYPE VARCHAR2(25), AMOUNT
NUMBER(20, 2))
tablespace TS_S
;

CREATE TABLE SUM_PROMO_SALES0 (PROMO_AMOUNT
NUMBER(20, 2))
tablespace TS_S
;

CREATE TABLE SUM_ALL_SALES0 (ALL_AMOUNT NUMBER(20,
2))
tablespace TS_S
;

INSERT INTO ALL_SALES0
SELECT
  P_TYPE,
  SUM(L_EXTENDEDPRI * (1-L_DISCOUNT))
FROM LINEITEM, PARTS
WHERE
  L_PARTKEY = P_PARTKEY
  AND L_SHIPDATE >= TO_DATE('1995-09-01','YYYY-MM-DD')
  AND L_SHIPDATE < ADD_MONTHS(TO_DATE('1995-09-01','YYYY-
MM-DD'), 1)
GROUP BY P_TYPE;

INSERT INTO SUM_PROMO_SALES0
SELECT
  SUM(AMOUNT)
FROM ALL_SALES0
WHERE
  TYPE LIKE 'PROMO%';

INSERT INTO SUM_ALL_SALES0
SELECT
  SUM(AMOUNT)
FROM ALL_SALES0;

SELECT
  100.00*PROMO_AMOUNT/ALL_AMOUNT AS PROMO_REVENUE
FROM SUM_PROMO_SALES0, SUM_ALL_SALES0;

DROP TABLE ALL_SALES0;

DROP TABLE SUM_PROMO_SALES0;

DROP TABLE SUM_ALL_SALES0;
-- Query 8 (Variant B)
SELECT

```

```

YEAR,
SUM(DECODE(NATION, 'BRAZIL', VOLUME, 0)) / SUM(VOLUME)
AS MKT_SHARE
FROM
(SELECT
  TO_CHAR(O_ORDERDATE, 'YYYY') AS YEAR,
  L_EXTENDEDPRICE*(1-L_DISCOUNT) AS VOLUME,
  N2.N_NAME AS NATION
FROM PARTS, SUPPLIER, LINEITEM, ORDERS, CUSTOMER,
NATION N1, NATION N2, REGION
WHERE
  P_PARTKEY = L_PARTKEY
  AND S_SUPPKEY = L_SUPPKEY
  AND L_ORDERKEY = O_ORDERKEY
  AND O_CUSTKEY = C_CUSTKEY
  AND C_NATIONKEY = N1.N_NATIONKEY
  AND N1.N_REGIONKEY = R_REGIONKEY
  AND R_NAME = 'AMERICA'
  AND S_NATIONKEY = N2.N_NATIONKEY
  AND O_ORDERDATE BETWEEN TO_DATE('1995-01-01', 'YYYY-
MM-DD') AND TO_DATE('1996-12-31', 'YYYY-MM-DD')
  AND P_TYPE = 'ECONOMY ANODIZED STEEL'
) ALL_NATIONS
GROUP BY YEAR
ORDER BY YEAR;
-- Query 12 (Variant B)
SELECT
  L_SHIPMODE,
  SUM(DECODE(O_ORDERPRIORITY, '1-URGENT', 1, '2-HIGH', 1, 0))
AS HIGH_LINE_COUNT,
  SUM(DECODE(O_ORDERPRIORITY, '1-URGENT', 0, '2-HIGH', 0, 1))
AS LOW_LINE_COUNT
FROM ORDERS, LINEITEM
WHERE
  O_ORDERKEY = L_ORDERKEY
  AND L_SHIPMODE IN ('MAIL', 'SHIP')
  AND L_COMMITDATE < L_RECEIPTDATE
  AND L_SHIPDATE < L_COMMITDATE
  AND L_RECEIPTDATE >= TO_DATE('1994-01-01', 'YYYY-MM-DD')
  AND L_RECEIPTDATE < ADD_MONTHS(TO_DATE('1994-01-01',
'YYYY-MM-DD'), 12)
GROUP BY L_SHIPMODE
ORDER BY L_SHIPMODE;
-- Query 17 (Original)
SELECT
  SUM(L_EXTENDEDPRICE)/7.0 AS AVG_YEARLY
FROM LINEITEM, PARTS
WHERE
  P_PARTKEY = L_PARTKEY
  AND P_BRAND = 'Brand#23'
  AND P_CONTAINER = 'MED BOX'
  AND L_QUANTITY <
  (SELECT
    0.2* AVG(L_QUANTITY)
  FROM LINEITEM
  WHERE
    L_PARTKEY = P_PARTKEY
  );
-- Query 3
set_fetchrows=10;
SELECT
  L_ORDERKEY,
  SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT)) AS REVENUE,
  TO_CHAR(O_ORDERDATE, 'YYYY-MM-DD'),
  O_SHIPPRIORITY
FROM CUSTOMER, ORDERS, LINEITEM
WHERE

```

```

  C_MKTSEGMENT = 'BUILDING'
  AND C_CUSTKEY = O_CUSTKEY
  AND L_ORDERKEY = O_ORDERKEY
  AND O_ORDERDATE < TO_DATE('1995-03-15', 'YYYY-MM-DD')
  AND L_SHIPDATE > TO_DATE('1995-03-15', 'YYYY-MM-DD')
GROUP BY L_ORDERKEY, O_ORDERDATE, O_SHIPPRIORITY
ORDER BY REVENUE DESC, O_ORDERDATE;
-- Query 5
SELECT
  N_NAME,
  SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT)) AS REVENUE
FROM CUSTOMER, ORDERS, LINEITEM, SUPPLIER, NATION,
REGION
WHERE
  C_CUSTKEY = O_CUSTKEY
  AND O_ORDERKEY = L_ORDERKEY
  AND L_SUPPKEY = S_SUPPKEY
  AND C_NATIONKEY = S_NATIONKEY
  AND S_NATIONKEY = N_NATIONKEY
  AND N_REGIONKEY = R_REGIONKEY
  AND R_NAME = 'ASIA'
  AND O_ORDERDATE >= TO_DATE('1994-01-01', 'YYYY-MM-DD')
  AND O_ORDERDATE < ADD_MONTHS(TO_DATE('1994-01-
01', 'YYYY-MM-DD'), 12)
GROUP BY N_NAME
ORDER BY REVENUE DESC;
-- Query 13 (Original)
SELECT
  YEAR,
  SUM(REVENUE) AS REVENUE
FROM
(SELECT
  TO_CHAR(O_ORDERDATE, 'YYYY') AS YEAR,
  L_EXTENDEDPRICE * (1-L_DISCOUNT) AS REVENUE
FROM LINEITEM, ORDERS
WHERE
  O_ORDERKEY = L_ORDERKEY
  AND O_CLERK = 'Clerk#000000088'
  AND L_RETURNFLAG = 'R'
) PERFORMANCE
GROUP BY YEAR
ORDER BY YEAR;
-- Query 7 (Original)
SELECT
  SUPP_NATION,
  CUST_NATION,
  YEAR,
  SUM(VOLUME) AS REVENUE
FROM
(SELECT
  N1.N_NAME AS SUPP_NATION,
  N2.N_NAME AS CUST_NATION,
  TO_CHAR(L_SHIPDATE, 'YYYY') AS YEAR,
  L_EXTENDEDPRICE * (1-L_DISCOUNT) AS VOLUME
FROM SUPPLIER, LINEITEM, ORDERS, CUSTOMER, NATION N1,
NATION N2
WHERE
  S_SUPPKEY = L_SUPPKEY
  AND O_ORDERKEY = L_ORDERKEY
  AND C_CUSTKEY = O_CUSTKEY
  AND S_NATIONKEY = N1.N_NATIONKEY
  AND C_NATIONKEY = N2.N_NATIONKEY
  AND ((N1.N_NAME = 'FRANCE' AND N2.N_NAME = 'GERMANY')
OR (N1.N_NAME = 'GERMANY' AND N2.N_NAME = 'FRANCE'))
  AND L_SHIPDATE BETWEEN TO_DATE('1995-01-01', 'YYYY-MM-
DD') AND TO_DATE('1996-12-31', 'YYYY-MM-DD')
) SHIPPING

```

GROUP BY SUPP_NATION, CUST_NATION, YEAR
ORDER BY SUPP_NATION, CUST_NATION, YEAR;

R	F	3779140.00	5328886172.99	5062370635.93
5265431221.82	25.55	36025.46	0.05	147920.00

-- Query 9 (Original)

```
SELECT
NATION,
YEAR,
SUM(AMOUNT) AS SUM_PROFIT
FROM
(SELECT
N_NAME AS NATION,
TO_CHAR(O_ORDERDATE,'YYYY') AS YEAR,
L_EXTENDEDPRI * (1-L_DISCOUNT) - PS_SUPPLYCOST *
L_QUANTITY AS AMOUNT
FROM PARTS, SUPPLIER, LINEITEM, PARTSUPP, ORDERS,
NATION
WHERE
S_SUPPKEY = L_SUPPKEY
AND PS_SUPPKEY = L_SUPPKEY
AND PS_PARTKEY = L_PARTKEY
AND P_PARTKEY = L_PARTKEY
AND O_ORDERKEY = L_ORDERKEY
AND S_NATIONKEY = N_NATIONKEY
AND P_NAME LIKE '%green%'
) PROFIT
GROUP BY NATION, YEAR
ORDER BY NATION, YEAR DESC;
```

4 rows processed.
Statement Processed in 4.80 seconds.

Ended Executing this Query at Mon Jun 15 12:58:45 1998

Query Started at 7597.62
Query Ended at 7602.42
Query Processed in 4.80 seconds.

-- Query 4 (Original)

Began Executing this Query at Mon Jun 15 12:58:45 1998

```
SELECT
O_ORDERPRIORITY,
COUNT(*) AS ORDER_COUNT
FROM ORDERS
WHERE
O_ORDERDATE >= TO_DATE('1993-07-01','YYYY-MM-DD')
AND O_ORDERDATE < ADD_MONTHS(TO_DATE('1993-07-
01','YYYY-MM-DD'),3)
AND EXISTS
(SELECT
*
FROM LINEITEM
WHERE
L_ORDERKEY = O_ORDERKEY
AND L_COMMITDATE < L_RECEIPTDATE
)
GROUP BY O_ORDERPRIORITY
ORDER BY O_ORDERPRIORITY
```

O_ORDERPRIORITY	ORDER_COUNT
1-URGENT	999.00
2-HIGH	1002.00
3-MEDIUM	1021.00
4-NOT SPECIFIED	997.00
5-LOW	1089.00

5 rows processed.
Statement Processed in 1.98 seconds.

Ended Executing this Query at Mon Jun 15 12:58:47 1998

Query Started at 7602.42
Query Ended at 7604.41
Query Processed in 1.98 seconds.

-- Query 15 (Variant B)

Began Executing this Query at Mon Jun 15 12:58:47 1998

Output from Validation Queries

Begin Execution at Mon Jun 15 12:58:40 1998

-- TPC-D Parameter Substitution (Version 1.2.0)
-- using default substitutions
-- Query 1 Original

```
SELECT
L_RETURNFLAG,
L_LINESTATUS,
SUM(L_QUANTITY) AS SUM_QTY,
SUM(L_EXTENDEDPRI) AS SUM_BASE_PRICE,
SUM(L_EXTENDEDPRI * (1 - L_DISCOUNT)) AS
SUM_DISC_PRICE,
SUM(L_EXTENDEDPRI * (1 - L_DISCOUNT) * (1 + L_TAX)) AS
SUM_CHARGE,
AVG(L_QUANTITY) AS AVG_QTY,
AVG(L_EXTENDEDPRI) AS AVG_PRICE,
AVG(L_DISCOUNT) AS AVG_DISC,
COUNT(*) AS COUNT_ORDER
FROM LINEITEM
WHERE
L_SHIPDATE <= TO_DATE('1998-12-01','YYYY-MM-DD') - 90
GROUP BY L_RETURNFLAG, L_LINESTATUS
ORDER BY L_RETURNFLAG, L_LINESTATUS

L_RETURNFLAG L_LINESTATUS SUM_QTY
SUM_BASE_PRICE SUM_DISC_PRICE
SUM_CHARGE AVG_QTY AVG_PRICE AVG_DISC
COUNT_ORDER
A F 3773034.00 5319329289.68 5053976845.78
5256336547.68 25.51 35964.01 0.05 147907.00
N F 100245.00 141459686.10 134380852.77
139710306.87 25.63 36160.45 0.05 3912.00
N O 7464940.00 10518546073.98 9992072944.46
10392414192.06 25.54 35990.13 0.05 292262.00
```

```
CREATE TABLE REVENUE0 (SUPPLIER_NO INTEGER,
TOTAL_REVENUE NUMBER(20, 2))
tablespace TS_S
```

Statement Processed in 0.03 seconds.

```
INSERT INTO REVENUE0
SELECT
L_SUPPKEY,
SUM(L_EXTENDEDPRI*(1-L_DISCOUNT))
FROM LINEITEM
WHERE
L_SHIPDATE >= TO_DATE('1996-01-01','YYYY-MM-DD')
AND L_SHIPDATE < ADD_MONTHS(TO_DATE('1996-01-01','YYYY-
MM-DD'),3)
GROUP BY L_SUPPKEY
Statement Processed in 0.53 seconds.
```

```
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
389 00      Supplier#000000389
PB1Lx0xx6LMz3h7Rx63m6j3QmMx      34-885-883-5717
1418538 21
```

1 row(s) processed.
Statement Processed in 0.01 seconds.

```
DROP TABLE REVENUE0
Statement Processed in 0.19 seconds.
Setting the number of rows to fetch to: 20
```

Ended Executing this Query at Mon Jun 15 12:58:48 1998

Query Started at 7604.41
Query Ended at 7605.16
Query Processed in 0.76 seconds.

-- Query 10

Began Executing this Query at Mon Jun 15 12:58:48 1998

```
SELECT
C_CUSTKEY,
C_NAME,
```

```
SUM(L_EXTENDEDPRI*(1-L_DISCOUNT)) AS REVENUE,
C_ACCTBAL,
N_NAME,
C_ADDRESS,
C_PHONE,
C_COMMENT
FROM CUSTOMER, ORDERS, LINEITEM, NATION
WHERE
C_CUSTKEY = O_CUSTKEY
AND L_ORDERKEY = O_ORDERKEY
AND O_ORDERDATE >= TO_DATE('1993-10-01','YYYY-MM-DD')
AND O_ORDERDATE < ADD_MONTHS(TO_DATE('1993-10-01',
'YYYY-MM-DD'), 3)
AND L_RETURNFLAG = 'R'
AND C_NATIONKEY = N_NATIONKEY
GROUP BY C_CUSTKEY, C_NAME, C_ACCTBAL, C_PHONE,
N_NAME, C_ADDRESS, C_COMMENT
ORDER BY REVENUE DESC
```

```
C_CUSTKEY    C_NAME    REVENUE
C_ACCTBAL
N_NAME    C_ADDRESS    C_PHONE
C_COMMENT
9722.00      Customer#000009722      464618.26      474.04
CANADA      I Mwzn4NAk6j      13-518-602-
8070
5L 50Oy RSgBAzPxmOSi5wk6xxOR7kh2nnPlgy7LBng2hOw5B01
RmCM120L24Pkg7PS1zwC11BCnz4L6i15PkixP26i66
12800.00      Customer#000012800      444265.64      1900.84
PERU      57zjB3CQx4P4OB2R2MBi2mwhSIIM4mn 4 nC6
27-142-205-3552
0hwglS77RB56Rx436IQ0N16CxbOPnmyhgwz
5z64wnj1kiC4jL350mM41y71hNxBllPjyA4hiN1wzjIM7SCxAN244mk2A
1025.00      Customer#000001025      442028.02      3363.46
INDIA      lkiSn154M5ROi      18-588-456-4616
0BI45z233Rniw0O0064nPBgP16kimO0y74iLh73g1N4 m310 jQ
yQzPA50iC 3MA75g2Bj162Nw4P
13028.00      Customer#000013028      441692.24      -452.66
UNITED KINGDOM      yP714ORSNgNN2LA3L5B
33-253-660-2127
xPkmnhL2BkhkNyww4khxwwAymN h11PSjBCNmi50LkyOhO6CC
5nzOQCALzliOk2R66w l05hRPO3iSP
3694.00      Customer#000003694      438180.07      2960.44
UNITED KINGDOM      2CcklmCBOCC      33-
421-331-3127
MzLxQxLiLx3MPxiAwg1B5kg61zxkPnk
xiAm6PhMMAA Q2nzN3S6zzgP
x70w0lhhPx4QRzIMMy0204AI3mBO7jh2jAPON60wg367z
976.00      Customer#000000976      435897.63      7772.85
ROMANIA      QzR 56Px1kgS wANnAz02RS 30n Pm
29-436-660-4732
kzn32776 gwzkMzzzO4yxOAnkR 7hR 4R 4x2SMwilz3x6h
nN7OnNLRMmI3 kz5SLwi1ykiOxiwS4g0wmA5A 4hmgBSwRRiQ1
8206.00      Customer#000008206      429905.11      6046.36
ARGENTINA      P yMg30BBBx NMgC03Amzn2
11-571-859-1370
hLi122RMPmLC36Oy0kxO71zz2wCR0QQCI7z26hiQ3mM
13532.00      Customer#000013532      427731.80      -924.18
KENYA
6ij7M5PBMx2kwwyz62Oj4SL5S0mRCwl3m1Rmw 24-525-332-7244
7ih7yRz214zO67AiNPx64nO5i5k yj6i3LA5PCL15Q4QIA3li60M1P
iBxCixg6 1hCh2RChjOzk5R OnO 1OhhC3m4631m5
12745.00      Customer#000012745      422327.69      9691.33
CHINA      SgS1LMC4gB2NM3wh      28-985-
189-6174
jI72wjSw0 S6 7L4Cgxw PkyO5Ni2LL7LBR
2344.00      Customer#000002344      411240.11      5597.22
```

```
MOROCCO          O3PC7ikBgw OAzPAlm2P
426zm3BnBN6Q1O 6N  25-593-745-7663
5NBn0wRNngLw2z5kyn1AhLOASyg6SMhM
i2kMOyxARAniO0Q5j4CBNARix7ABIMAC
2656.00      Customer#000002656      401185.95      8115.55
ALGERIA          On55IAS3Rm5RxS m      10-667-
469-8092
46ABx4jgni mMBMPLxRhYPQM4RNS 5yO1L7zSOmk
MhPxAXQQ6lQnLj
17LymOhi4i5innzOyB2OlXzmw3gmX0SxiyBN5CSMNgCkLCKMgO
59.00      Customer#000000059      400759.15      3458.60
ARGENTINA          wP6CMYClly0IS4CAM1mzm      11-
355-584-3112
l1g7xBCxxC7SM 5AkmmAk00677O1MZA2R7A0C0x0Njixj56jL2IN
PNkSNQiy55m6ki3OgnhM47mSR7B
7069.00      Customer#000007069      396217.52      8198.94
INDONESIA          55Cw7ChL4Bi5ONn2A4m2i2n4nSNQQMjml
19-644-744-1798
6jNS624l75zlxNli4lxO5zyPykPS1xnliS0NhkgOAKSx7P
6553.00      Customer#000006553      385863.59      8985.90
MOZAMBIQUE          R3LnnxONBJCLC0MRkxy7      26-
166-724-4677
S7CkNLwA3kh006j7l1wAIC25Bw6AMQ6i 6C0OSS6O7ARNNny60Ogh
3642mRxyiAgy5yk 3nPO4473wkNg5R6gzO4lz3zmM2m7MlAILCC
3095.00      Customer#000003095      384246.11      8829.21
IRAQ          S1gMCnBLwzi mCgB664 jI00LI1Snh1iPMgCgR5
21-847-218-8188
3LSx7PxS
A4A5Cl3gAy3mg4Qj2xQlyx7xM1kA664AM7zmMmzORh3C1h
MO3nw6MymiljAMg65hOMB4Sn44kO w0lin7
3391.00      Customer#000003391      382541.78      7742.35
CANADA          m3 CORmQNLZkShymLS iMkCimRSI20 NB
13-592-494-2668
ynMlmhMBA5ikC1nCghImAhQ0 675S3y2R33yjkNPQOS
13678.00      Customer#000013678      376280.56      9030.40
MOROCCO          BMk77lQm1lwNAOLghAk3hCwNI4
25-306-951-3937
mOS55RASx1wP136nQ5xBLznLhgw1kQ6PO6imNxQ7kR0x71P0SzBy
Mzh
6062.00      Customer#000006062      374512.65      1370.35
CANADA          n5zzil60zyxAlkzx7x1nihigPzR OBkR znMOMh
13-756-700-4918
4zAm4wNB
li4QRPgPz2wM541x043hmLj4Q3LBALCP16hj2RQB01OMNly7ww1Q
P7w5i SSn0jNhAR yQmmz1hi5j3
554.00      Customer#000000554      373004.47      8395.57
BRAZIL          jC5zhQky4zQB27IB5Sm AqhQ Px0      12-
938-503-7317
OnxCl3 xSmILQO 1M
2n0NCIRInMMxP25j26x2igLhNOxjgMgmwwwy7OkjzCACOG0z2LAjOm0
RPRmOPICAAQwLIQsg 1yS3 gLCM1M2BzinSjPI3nwAkk
13126.00      Customer#000013126      371722.00      6172.91
INDIA          xPAS4MnPh40i5Q2h4NQ61zz4RkyAWANA
18-288-190-4145
nniMkAN6C0CIQ0mMmPz27liz4hk6L
2MlwPxh42N1lOR2hRwxzlwMkxO4MAyz7RCj43NxLwQ3m6P27yAj
```

20 rows processed.

Statement Processed in 3.71 seconds.

Ended Executing this Query at Mon Jun 15 12:58:51 1998

Query Started at 7605.16

Query Ended at 7608.87

Query Processed in 3.71 seconds.

-- Query 11 (Variant A)

Began Executing this Query at Mon Jun 15 12:58:51 1998

```
CREATE TABLE PART_VALUE0 (PARTNO INTEGER, VALUE
NUMBER(20, 2) )
tablespace TS_S
```

Statement Processed in 0.11 seconds.

```
INSERT INTO PART_VALUE0
SELECT
PS_PARTKEY,
SUM(PS_SUPPLYCOST*PS_AVAILQTY)
FROM PARTSUPP, SUPPLIER, NATION
WHERE
PS_SUPPKEY = S_SUPPKEY
AND S_NATIONKEY = N_NATIONKEY
AND N_NAME = 'GERMANY'
GROUP BY PS_PARTKEY
Statement Processed in 0.45 seconds.
```

```
CREATE TABLE SUM_PART_VALUE0 (TOTAL_VALUE
NUMBER(20, 2))
tablespace TS_S
```

Statement Processed in 0.03 seconds.

```
INSERT INTO SUM_PART_VALUE0
SELECT
SUM(PS_SUPPLYCOST*PS_AVAILQTY) * 0.0010000000
FROM PARTSUPP, SUPPLIER, NATION
WHERE
PS_SUPPKEY = S_SUPPKEY
AND S_NATIONKEY = N_NATIONKEY
AND N_NAME = 'GERMANY'
Statement Processed in 2.11 seconds.
```

```
SELECT
PARTNO AS P_PARTKEY,
VALUE
FROM PART_VALUE0
WHERE
VALUE >
(SELECT
SUM(TOTAL_VALUE)
FROM SUM_PART_VALUE0
)
ORDER BY VALUE DESC
```

P_PARTKEY	VALUE
12098.00	16227681.21
5134.00	15709338.52
13334.00	15023662.41
17052.00	14351644.20
3452.00	14070870.14
12552.00	13332469.18
1084.00	13170428.29
5797.00	13038622.72
12633.00	12892561.61
403.00	12856217.34
1833.00	12024581.72
2084.00	11502875.36
17349.00	11354213.05

18427.00 11282385.24
2860.00 11262529.95
17852.00 10934711.93
9871.00 10889253.68
12231.00 10841131.39
6366.00 10759786.81
12146.00 10257362.66
5043.00 10226395.88
12969.00 10125777.93

22 rows processed.
Statement Processed in 0.02 seconds.

DROP TABLE PART_VALUE0
Statement Processed in 0.04 seconds.

DROP TABLE SUM_PART_VALUE0
Statement Processed in 0.04 seconds.

Ended Executing this Query at Mon Jun 15 12:58:54 1998

Query Started at 7608.87
Query Ended at 7611.68
Query Processed in 2.80 seconds.

-- Query 6

Began Executing this Query at Mon Jun 15 12:58:54 1998

SELECT
SUM(L_EXTENDEDPRI * L_DISCOUNT) AS REVENUE
FROM LINEITEM
WHERE
L_SHIPDATE >= TO_DATE('1994-01-01','YYYY-MM-DD')
AND L_SHIPDATE < ADD_MONTHS(TO_DATE('1994-01-01','YYYY-MM-DD'),12)
AND L_DISCOUNT BETWEEN .06 - 0.01 AND .06 + 0.01
AND L_QUANTITY < 24

REVENUE
11450588.04

1 row processed.
Statement Processed in 0.51 seconds.
Setting the number of rows to fetch to: 100

Ended Executing this Query at Mon Jun 15 12:58:55 1998

Query Started at 7611.68
Query Ended at 7612.18
Query Processed in 0.51 seconds.

-- Query 2 Variant A

Began Executing this Query at Mon Jun 15 12:58:55 1998

SELECT
S_ACCTBAL,
S_NAME,
N_NAME,
P_PARTKEY,
P_MFGR,
S_ADDRESS,
S_PHONE,
S_COMMENT
FROM PARTS, SUPPLIER, PARTSUPP, NATION, REGION
WHERE
P_PARTKEY = PS_PARTKEY
AND S_SUPPKEY = PS_SUPPKEY
AND P_SIZE = 15
AND P_TYPE LIKE '%BRASS'
AND S_NATIONKEY = N_NATIONKEY
AND N_REGIONKEY = R_REGIONKEY
AND R_NAME = 'EUROPE'
AND (P_PARTKEY,PS_SUPPLYCOST) IN
(SELECT
PS_PARTKEY,
MIN(PS_SUPPLYCOST)
FROM PARTSUPP, SUPPLIER, NATION, REGION
WHERE
S_SUPPKEY = PS_SUPPKEY
AND S_NATIONKEY = N_NATIONKEY
AND N_REGIONKEY = R_REGIONKEY
AND R_NAME = 'EUROPE'
GROUP BY PS_PARTKEY
)
ORDER BY S_ACCTBAL DESC, N_NAME, S_NAME, P_PARTKEY

S_ACCTBAL	S_NAME	N_NAME
P_PARTKEY	P_MFGR	
S_ADDRESS		S_PHONE
S_COMMENT		
9828.21	Supplier#000000647	UNITED KINGDOM
13120.00	Manufacturer#5	
jB16PyPyB7B152MjSPw3mS		33-258-202-4782
z1QhSiMj11Bm7C0Lwh6QI0B1R2Mg4CLn		
LhiP0wIMzy72hIkP715in2y6RS6N130iz51nSRL5gOg5S26hPCCQN2L		
9508.37	Supplier#000000070	FRANCE
3563.00	Manufacturer#1	
M5C616R5h5SIMR3zzmLkSw24j2		16-821-608-1166
m7z0CPShmBkhlChBAi3LkQ2CLw		
mhl6QP362RPS3044CB2y41yhOhjIBin0CL7yhxmhS4hBM07kQ1yyjQjz3C		
9508.37	Supplier#000000070	FRANCE
17268.00	Manufacturer#4	
M5C616R5h5SIMR3zzmLkSw24j2		16-821-608-1166
m7z0CPShmBkhlChBAi3LkQ2CLw		
mhl6QP362RPS3044CB2y41yhOhjIBin0CL7yhxmhS4hBM07kQ1yyjQjz3C		
9453.01	Supplier#000000802	ROMANIA
10021.00	Manufacturer#5	
5yARQNSLNRAlOlBnkNQCik3SOlyCik7nmRhA2h0		29-342-882-6463
65y3RQ2i0OP6Nz7mS hC		
PxwLy7L1jQy6OI63xO3BCz52Rm1zm0MziCMLij2n6wky51mBOWx		
Qh52iz QB1545Amxyj		
9453.01	Supplier#000000802	ROMANIA
13275.00	Manufacturer#4	
5yARQNSLNRAlOlBnkNQCik3SOlyCik7nmRhA2h0		29-342-882-6463
65y3RQ2i0OP6Nz7mS hC		
PxwLy7L1jQy6OI63xO3BCz52Rm1zm0MziCMLij2n6wky51mBOWx		
Qh52iz QB1545Amxyj		

9192.10 Supplier#000000115 UNITED KINGDOM
 13325.00 Manufacturer#1
 h0m3lzlSPMw2B0ny7LNYNCkjRRn7iyMILBLA 33-597-248-1220
 1QzQjhSyx
 ixm2lgz2Ry7075RL3MS5236x56hxmR0wLNL0LBxm164LzCmMALzOAJ
 n4kz7i4wjOICONI1C51M7nCMx66SBRAQA
 9032.15 Supplier#000000959 GERMANY
 4958.00 Manufacturer#4
 205LNCzxMCnQ5gnz4n S3ynP6Mhnw 17-108-642-3106
 Px z7kOx56l7jQz NwBBQhky yM7kLgxRQw5zw6 426Bm55IC6
 OkQ7hQPLixjM7y47BNP16CRi0kjk354lgxh
 8702.02 Supplier#000000333 RUSSIA
 11810.00 Manufacturer#3
 5iWkgN5n2BN15OmQk2602h0N6NzxPyiPN5inj 32-508-202-6136
 SgimAjmn3wL7RlXmh3LCwOPnhjyl 7xxzxAN 4ACx43y65NwQ7P
 8615.50 Supplier#000000812 FRANCE
 10551.00 Manufacturer#2
 h4i2M2O0 ky1g2mIBOmXjzj0hA2h6nkSNhP 16-585-724-6633
 57i0NAyR0RP2jOh54C6B22OISL
 8615.50 Supplier#000000812 FRANCE
 13811.00 Manufacturer#4
 h4i2M2O0 ky1g2mIBOmXjzj0hA2h6nkSNhP 16-585-724-6633
 57i0NAyR0RP2jOh54C6B22OISL
 8488.53 Supplier#000000367 RUSSIA
 6854.00 Manufacturer#4
 nkmQ2Qzgh0wA 3x Sn2S7N5gmSOj xwC COSn6 32-458-198-9557
 35C2RROP C Nlgi2N
 SxAj0hQkn7kP5z4wSxSwgMxj6k4MRmh0S2Qm7R3z4JB OOQBMI
 8430.52 Supplier#000000646 FRANCE
 11384.00 Manufacturer#3
 61SjP6S y B0 32i11 16-601-220-5489
 kiw4NSNBnxy5kywzwyx0PMM21xiMOhxR423Akkm
 Q7CNwRzQS23Nzz22 mnm6P377Q3Mj7n 56BLm6lxwllh kSmN
 8271.39 Supplier#000000146 RUSSIA
 4637.00 Manufacturer#5
 wh yPSk6hNBIB4i33iQ0wS0 RhBhQ4zQ3lz 32-792-619-3155
 jiwgwliRO63 n7OM2MP0hg3L1mLwBMLmMIS4Cgyn
 LA5PwC2P0AS6g3C5mkQjO72NPig731m
 8096.98 Supplier#000000574 RUSSIA
 323.00 Manufacturer#4
 hCQj4Cgx43xx jgP4QkL7gLN65 32-866-246-8752
 OhxNj6SIB56315B3k5SCBzwQyLk76zlj4Ow2Q
 BC2wACKxh3S0RCyx6nARzSQR2010k0BCPhOg6yQm
 7392.78 Supplier#000000170 UNITED KINGDOM
 7655.00 Manufacturer#2
 PCxijzNQihLNxgLw0SiMmQ 33-803-340-5398
 M116S1xzg5i4C3k7OPLQi3Cimhghz2BCIQk
 g5Ag12QSBhglANnw4MR MBS 72A
 7205.20 Supplier#000000477 GERMANY
 10956.00 Manufacturer#5
 Mimj6403h zmAzAgg Bjy05O 2z 17-180-144-7991
 yRlyR SnMxmhpJAmbw
 S02AxQ6yOhBRIOWzmlxzO0A2Sx075kjlAkn7z2
 O0S7hy0BiknwOQm6Pmz3gL4gj2z7
 6820.35 Supplier#000000007 UNITED KINGDOM
 13217.00 Manufacturer#5
 z45m2jBRzli5ILNz4 33-990-965-2201
 1PhngjmiSQl0RzRACP0i4S70xSL
 QPSBM16072SkMLCgm4O0MjARLNQk3g1P3BB32AgBMI462B0CP7
 Rh24
 6721.70 Supplier#000000954 FRANCE
 4191.00 Manufacturer#3
 OM7xnNxnkgQ mzh2g3RQmg1g 16-537-341-8517
 5ni3yCkmz5ymx0kCg74zhLA B516S1w152AkiByx1NI
 NgghAkkmNz1jASj4mxzxznOySg7hAyM3MRRnBj
 6329.90 Supplier#000000996 GERMANY
 10735.00 Manufacturer#2
 k6135gA3zPwNi7L3Ri45mInACjngOQQBB300jyA 17-447-811-3282

PBO7wjlQMm1h3AAA 1NQAI0kkjnkRNQ0 mh1z6QSOgC5IP1
 ykmzNR20OIN506ARS0z3j
 6173.87 Supplier#000000408 RUSSIA
 18139.00 Manufacturer#1
 Cni6 zR5C4h104POx5h05 mg53CQ2Sw4SAM2M2x 32-858-724-2950
 10SxMOWhjON3khzQ124gNnyw7B4nL7m14L5ISR
 5364.99 Supplier#000000785 RUSSIA
 13784.00 Manufacturer#4
 71OnPzQkC2P1hRNRgjyQP4n1 32-297-653-2203
 kiiPQ3ik7R ykAhRx43Rw70L1Ok
 7AMi3AjRw7klwxwyiL6S2O1COyS4QB46m5M167mjMwCm0w
 5069.27 Supplier#000000328 GERMANY
 16327.00 Manufacturer#1
 5O4033xSgml 17-231-513-5721
 OMk3ALAPNmj0BLMAS7M1nCAs
 4xLj5i1y2klx3nP126gAxPgANmk6zSi6 3A7m I11BOWiC6xLB4hBRIPM
 4941.88 Supplier#000000321 ROMANIA
 7320.00 Manufacturer#5
 hYLQ mg42S2kAMlj M3BwMSjS 29-573-279-1406
 y2644kMhOkPCm5P5y7Lmz7OR6mgSmBN631RggmC
 4672.25 Supplier#000000239 RUSSIA
 12238.00 Manufacturer#1
 y4ymj7B5BN1nMSkwPPggAl 32-396-654-6826
 Py3RA2gykmSCmj0z3i7Rxzhz6OyR RxS C3S23LPQ
 4586.49 Supplier#000000680 RUSSIA
 5679.00 Manufacturer#3
 BP1NIw5nPMxRnOAwm 32-522-382-1620
 kAOy25RNO1Al im7SyIPzSym3M5OS52i6S576kn0S2k
 OmPBLIAzL6Ax7CM6iNi4CgCy6Bln7hlhxmLRng
 4518.31 Supplier#000000149 FRANCE
 18344.00 Manufacturer#5
 4B QSY5BI2 16-660-553-2456
 hijkPhgL1g4L1Q27y0Q42wh0Qz3PiL4NgkM4NNg1
 lIQ1yNNBk1C1QnIR07 4ki
 4315.15 Supplier#000000509 FRANCE
 18972.00 Manufacturer#2
 B5 iPRn7L4yMllgwCnRPMA 16-298-154-3365
 ygiPh7ymP7jBznmR2iQLLgmi1wik
 3526.53 Supplier#000000553 FRANCE
 8036.00 Manufacturer#4
 yLiOx2gMw 5iBi6AiNL60Q 16-599-552-3755
 L3ggShiRlyxmR4MNI7Rw7OQign6yO
 3526.53 Supplier#000000553 FRANCE
 17018.00 Manufacturer#3
 yLiOx2gMw 5iBi6AiNL60Q 16-599-552-3755
 L3ggShiRlyxmR4MNI7Rw7OQign6yO
 3294.68 Supplier#000000350 GERMANY
 4841.00 Manufacturer#4
 x5kRL2z1BPg0 BO 2hi1iOyh 30RRg0OPj 17-113-181-4017
 BjQznni44OmQ7S16yl3zxk2M6nM4M 27yMPML
 2972.26 Supplier#000000016 RUSSIA
 1015.00 Manufacturer#4
 B7wLkSLRjNS MS1C 32-822-502-4215
 C7w6S6QzhAPQmMmNMn1hA0iQOA 00m1NmC25wyQ461SA
 jy03zmRh22MLM00zhmi
 2963.09 Supplier#000000840 ROMANIA
 3080.00 Manufacturer#2
 1ynwQkNh0 CMRRCK4i306M2i0jykg6QNgSCAzy 29-781-337-5584
 7NRMx43Rm QjxML6hXlyN75LzxBwB0wSLx3
 S3Cwh52S6iLSOLhQm0 6C1 yzx3Pm6Sjg
 5By0BCPwOR32i1CQgxR0gB43gh
 2221.25 Supplier#000000771 ROMANIA
 13981.00 Manufacturer#2
 LAjCRj3nAMzzhmw0Sx1Mg 29-986-304-9006
 jhk0N7NhS23iCngC52BBC
 QjiLMowByx0LB5R070R2iCx113iQis7xNhBRA0xknlNxLiA
 1381.97 Supplier#000000104 FRANCE
 18103.00 Manufacturer#3

i QnI4 1 jiwM C2yxAyIL5R4SBQh54N6 16-434-972-6922
MwnBw1g71Pig2Am7nz0Mm5SNI7OwQLAkN56ji
906.07 Supplier#000000138 ROMANIA
8363.00 Manufacturer#4
1iBxSxL11Mh3 6LS6PILPNiIMjCQh22z6n5 29-533-434-6776
nLjQAmCw77R2jRMgz5LSyyx1QN I
4jMMO3RAkxOkzRmwQl3Qm5236k72RRPnim0 BkzQnBMM6A
PMmI2n
765.69 Supplier#000000799 RUSSIA
11276.00 Manufacturer#2
Am7yihz47mg NkgQL w By4 32-579-339-1495
MMRPNQ 4i66mQQPNniAiIL0PQ2C4yyBRn1nRlxnxkj5Ak45Pw
mQk1ROhZ66BRQiIL gPRQRy 56MyQ nS1N14R 7Ml6xhl2IOS3
727.89 Supplier#000000470 ROMANIA
6213.00 Manufacturer#3
gAySBM2N7 PgwP5kiP4n7BzOik0M 29-165-289-1523
zCkPgn
6wN5A3R47gijlQ3hNSLShP2RALxCiinkOy4wCwA1LCIBO5yiSC yBAA
lii
683.07 Supplier#000000651 RUSSIA
4888.00 Manufacturer#4
ymQ6PByCh4lzxBBPLB2wwOhRh47wQMQSPL 32-181-426-4490
kx6jhQkwzg6RkrGPLPM30Bgl1R726l1m5AMk0MmMQBQ
nCihLxhMgCgRih6MmMx0PglRQ7A Qnl72g50
167.56 Supplier#000000290 FRANCE
2037.00 Manufacturer#1
xm6n30QmBn75QPh7Py01klnB4n1lNh MBi1 16-675-286-5102
gi6y41Bg5
AkhmCh30Qyxx515xLi6MRjCQ4n2xjML7N0PNSCyBPwS1C
91.39 Supplier#000000949 UNITED KINGDOM
9430.00 Manufacturer#2
zCmkwm43LQ62Q3O 4nkNA 33-332-697-2768
Nm5yn23SxgPAMRz6B5CjAj04nkNx5IA7O
-314.06 Supplier#000000510 ROMANIA
17242.00 Manufacturer#4
hwx1kP6nQ0NgPO232nxAwL4QPjRN i64jOiPjC0 29-207-852-3454
5NAyRmwSCQ5hMB6RiN0Qyhghnl0BPjPINIP725AAOCw
-820.89 Supplier#000000409 GERMANY
2156.00 Manufacturer#5
Nk4z1lg25gNwMwO 2BnMon1P3k0yA7i2l 17-719-517-9836
CmLnngQywQ4hxx3266yz5Qgj
nOm1P1RC3gxmg4kmnNCSnM4PBO4hNQA1xyzC2LCQBP07w2z3
-845.44 Supplier#000000704 ROMANIA
9926.00 Manufacturer#5
zi jjmO4A57 mQgLmhALCM3B zL 29-300-896-5991
OSh0PRm6ByyS2wLBNPQPPI2CL0zN0Px1LNOlw6y0Rgij zP5hPCz
745nn6zhPORI6i7SynBCC4Qy4Qkn1 I
-942.73 Supplier#000000563 GERMANY
5797.00 Manufacturer#1
xSLly 4jm17OC 17-108-537-2691
1gSol3A0jnkAmQLlmygzmg7hhjAmRxN5SwjxM16Ok61 wm5N42

44 rows processed.

Statement Processed in 1.29 seconds.

Ended Executing this Query at Mon Jun 15 12:58:56 1998

Query Started at 7612.18

Query Ended at 7613.47

Query Processed in 1.29 seconds.

-- Query 16

Began Executing this Query at Mon Jun 15 12:58:56 1998

```
SELECT
P_BRAND,
P_TYPE,
P_SIZE,
COUNT(DISTINCT PS_SUPPKEY) AS SUPPLIER_CNT
FROM PARTSUPP,PARTS
WHERE
P_PARTKEY = PS_PARTKEY
AND P_BRAND <> 'Brand#45'
AND P_TYPE NOT LIKE 'MEDIUM POLISHED%'
AND P_SIZE IN (49, 14, 23, 45, 19, 3, 36, 9)
AND PS_SUPPKEY NOT IN
(SELECT
S_SUPPKEY
FROM SUPPLIER
WHERE
S_COMMENT LIKE '%Better Business Bureau%Complaints%'
)
GROUP BY P_BRAND, P_TYPE, P_SIZE
ORDER BY SUPPLIER_CNT DESC, P_BRAND, P_TYPE, P_SIZE
```

P_BRAND	P_TYPE	P_SIZE	SUPPLIER_CNT
Brand#14	SMALL ANODIZED NICKEL	45.00	12.00
Brand#22	SMALL BURNISHED BRASS	19.00	12.00
Brand#25	PROMO POLISHED COPPER	14.00	12.00
Brand#35	LARGE ANODIZED STEEL	45.00	12.00
Brand#35	PROMO BRUSHED COPPER	9.00	12.00
Brand#51	ECONOMY ANODIZED STEEL	9.00	12.00
Brand#53	LARGE BRUSHED NICKEL	45.00	12.00
Brand#11	ECONOMY POLISHED COPPER	14.00	8.00
Brand#11	LARGE PLATED STEEL	23.00	8.00
Brand#11	PROMO POLISHED STEEL	23.00	8.00
Brand#11	STANDARD ANODIZED COPPER	9.00	8.00
Brand#12	ECONOMY BURNISHED BRASS	9.00	8.00
Brand#12	LARGE ANODIZED BRASS	14.00	8.00
Brand#12	SMALL ANODIZED TIN	23.00	8.00
Brand#12	SMALL BRUSHED NICKEL	23.00	8.00
Brand#12	STANDARD ANODIZED BRASS	3.00	8.00
Brand#12	STANDARD BURNISHED TIN	23.00	8.00
Brand#13	ECONOMY POLISHED BRASS	9.00	8.00
Brand#13	LARGE BURNISHED COPPER	45.00	8.00
Brand#13	MEDIUM ANODIZED STEEL	23.00	8.00
Brand#13	MEDIUM PLATED NICKEL	3.00	8.00
Brand#13	PROMO BURNISHED BRASS	9.00	8.00
Brand#13	PROMO POLISHED BRASS	3.00	8.00
Brand#13	PROMO POLISHED TIN	36.00	8.00
Brand#13	SMALL BURNISHED STEEL	23.00	8.00
Brand#13	STANDARD BRUSHED STEEL	9.00	8.00
Brand#14	ECONOMY BRUSHED TIN	3.00	8.00
Brand#14	ECONOMY BURNISHED TIN	23.00	8.00
Brand#14	PROMO BRUSHED STEEL	9.00	8.00
Brand#14	PROMO PLATED TIN	45.00	8.00
Brand#15	ECONOMY PLATED TIN	9.00	8.00
Brand#15	STANDARD BRUSHED COPPER	14.00	8.00
Brand#15	STANDARD PLATED TIN	3.00	8.00
Brand#21	ECONOMY POLISHED TIN	3.00	8.00
Brand#21	PROMO POLISHED COPPER	9.00	8.00
Brand#21	PROMO POLISHED TIN	49.00	8.00
Brand#21	STANDARD PLATED BRASS	49.00	8.00
Brand#21	STANDARD PLATED NICKEL	49.00	8.00
Brand#22	ECONOMY ANODIZED TIN	49.00	8.00
Brand#22	ECONOMY BRUSHED BRASS	14.00	8.00
Brand#22	LARGE BURNISHED TIN	36.00	8.00

Brand#22	MEDIUM ANODIZED STEEL	36.00	8.00	Brand#51	SMALL BRUSHED COPPER	45.00	8.00
Brand#22	MEDIUM PLATED STEEL	9.00	8.00	Brand#51	SMALL BRUSHED TIN	36.00	8.00
Brand#22	PROMO POLISHED NICKEL	9.00	8.00	Brand#51	STANDARD POLISHED TIN	3.00	8.00
Brand#22	SMALL ANODIZED STEEL	19.00	8.00	Brand#52	ECONOMY ANODIZED STEEL	3.00	8.00
Brand#22	STANDARD ANODIZED COPPER	23.00	8.00	Brand#52	ECONOMY PLATED TIN	19.00	8.00
Brand#23	ECONOMY BRUSHED NICKEL	23.00	8.00	Brand#52	LARGE PLATED TIN	3.00	8.00
Brand#23	LARGE ANODIZED BRASS	9.00	8.00	Brand#52	MEDIUM ANODIZED TIN	19.00	8.00
Brand#23	LARGE ANODIZED STEEL	23.00	8.00	Brand#52	MEDIUM BURNISHED COPPER	3.00	8.00
Brand#23	SMALL BRUSHED COPPER	23.00	8.00	Brand#52	PROMO POLISHED BRASS	23.00	8.00
Brand#23	STANDARD BRUSHED TIN	3.00	8.00	Brand#52	SMALL PLATED COPPER	36.00	8.00
Brand#23	STANDARD BURNISHED NICKEL	49.00	8.00	Brand#52	SMALL POLISHED NICKEL	9.00	8.00
Brand#23	STANDARD PLATED NICKEL	36.00	8.00	Brand#52	STANDARD POLISHED NICKEL	45.00	8.00
Brand#24	ECONOMY ANODIZED BRASS	19.00	8.00	Brand#53	ECONOMY POLISHED STEEL	45.00	8.00
Brand#24	ECONOMY POLISHED BRASS	36.00	8.00	Brand#53	LARGE POLISHED NICKEL	3.00	8.00
Brand#24	LARGE BURNISHED STEEL	14.00	8.00	Brand#53	SMALL BRUSHED COPPER	14.00	8.00
Brand#24	MEDIUM PLATED NICKEL	36.00	8.00	Brand#53	STANDARD PLATED STEEL	45.00	8.00
Brand#25	ECONOMY BRUSHED STEEL	49.00	8.00	Brand#54	ECONOMY POLISHED BRASS	49.00	8.00
Brand#25	MEDIUM BURNISHED TIN	3.00	8.00	Brand#54	ECONOMY POLISHED TIN	23.00	8.00
Brand#25	PROMO ANODIZED TIN	36.00	8.00	Brand#54	MEDIUM BRUSHED STEEL	9.00	8.00
Brand#25	PROMO PLATED NICKEL	3.00	8.00	Brand#54	SMALL BURNISHED NICKEL	14.00	8.00
Brand#25	SMALL BURNISHED BRASS	3.00	8.00	Brand#54	SMALL PLATED TIN	14.00	8.00
Brand#31	LARGE ANODIZED BRASS	3.00	8.00	Brand#54	STANDARD BURNISHED STEEL	14.00	8.00
Brand#31	SMALL ANODIZED COPPER	3.00	8.00	Brand#54	STANDARD PLATED BRASS	23.00	8.00
Brand#31	SMALL ANODIZED NICKEL	9.00	8.00	Brand#55	MEDIUM BURNISHED TIN	36.00	8.00
Brand#31	SMALL ANODIZED STEEL	14.00	8.00	Brand#55	PROMO ANODIZED BRASS	14.00	8.00
Brand#32	MEDIUM ANODIZED STEEL	49.00	8.00	Brand#55	STANDARD BURNISHED COPPER	45.00	8.00
Brand#32	MEDIUM BURNISHED COPPER	19.00	8.00	Brand#15	STANDARD PLATED TIN	36.00	7.00
Brand#32	SMALL BURNISHED STEEL	23.00	8.00	Brand#21	SMALL POLISHED STEEL	3.00	7.00
Brand#32	STANDARD BURNISHED STEEL	45.00	8.00	Brand#23	SMALL POLISHED BRASS	49.00	7.00
Brand#34	ECONOMY ANODIZED NICKEL	49.00	8.00	Brand#34	MEDIUM BURNISHED NICKEL	3.00	7.00
Brand#34	LARGE BURNISHED TIN	49.00	8.00	Brand#42	STANDARD PLATED COPPER	19.00	7.00
Brand#34	PROMO ANODIZED TIN	3.00	8.00	Brand#51	LARGE POLISHED NICKEL	14.00	7.00
Brand#34	SMALL BRUSHED TIN	3.00	8.00	Brand#54	LARGE ANODIZED NICKEL	49.00	7.00
Brand#34	STANDARD BURNISHED TIN	23.00	8.00	Brand#11	ECONOMY ANODIZED BRASS	19.00	4.00
Brand#35	MEDIUM BRUSHED STEEL	45.00	8.00	Brand#11	ECONOMY ANODIZED BRASS	45.00	4.00
Brand#35	PROMO BURNISHED STEEL	14.00	8.00	Brand#11	ECONOMY ANODIZED NICKEL	36.00	4.00
Brand#35	SMALL BURNISHED STEEL	23.00	8.00	Brand#11	ECONOMY BRUSHED COPPER	3.00	4.00
Brand#35	SMALL POLISHED COPPER	14.00	8.00	Brand#11	ECONOMY BRUSHED COPPER	9.00	4.00
Brand#35	STANDARD PLATED COPPER	9.00	8.00	Brand#11	ECONOMY BRUSHED STEEL	9.00	4.00
Brand#41	ECONOMY BRUSHED BRASS	23.00	8.00	Brand#11	ECONOMY BRUSHED STEEL	36.00	4.00
Brand#41	LARGE BURNISHED STEEL	23.00	8.00	Brand#11	ECONOMY BURNISHED BRASS	36.00	4.00
Brand#41	PROMO BURNISHED TIN	14.00	8.00	Brand#11	ECONOMY BURNISHED COPPER	9.00	4.00
Brand#41	PROMO PLATED STEEL	36.00	8.00	Brand#11	ECONOMY BURNISHED COPPER	49.00	4.00
Brand#41	PROMO POLISHED TIN	19.00	8.00	Brand#11	ECONOMY BURNISHED NICKEL	14.00	4.00
Brand#41	SMALL BURNISHED COPPER	23.00	8.00	Brand#11	ECONOMY BURNISHED NICKEL	49.00	4.00
Brand#42	LARGE POLISHED TIN	14.00	8.00	Brand#11	ECONOMY PLATED COPPER	19.00	4.00
Brand#42	MEDIUM ANODIZED TIN	49.00	8.00	Brand#11	ECONOMY PLATED NICKEL	45.00	4.00
Brand#42	MEDIUM BRUSHED TIN	14.00	8.00	Brand#11	ECONOMY PLATED TIN	9.00	4.00
Brand#42	MEDIUM BURNISHED NICKEL	23.00	8.00	Brand#11	ECONOMY POLISHED BRASS	3.00	4.00
Brand#42	MEDIUM PLATED COPPER	45.00	8.00	Brand#11	ECONOMY POLISHED COPPER	3.00	4.00
Brand#42	MEDIUM PLATED TIN	45.00	8.00	Brand#11	ECONOMY POLISHED COPPER	45.00	4.00
Brand#42	SMALL PLATED COPPER	36.00	8.00	Brand#11	ECONOMY POLISHED NICKEL	36.00	4.00
Brand#43	ECONOMY BRUSHED STEEL	45.00	8.00	Brand#11	ECONOMY POLISHED STEEL	23.00	4.00
Brand#43	LARGE BRUSHED COPPER	19.00	8.00	Brand#11	ECONOMY POLISHED TIN	14.00	4.00
Brand#43	PROMO BRUSHED BRASS	36.00	8.00	Brand#11	LARGE ANODIZED COPPER	23.00	4.00
Brand#43	SMALL BURNISHED TIN	45.00	8.00	Brand#11	LARGE ANODIZED NICKEL	9.00	4.00
Brand#43	SMALL PLATED COPPER	45.00	8.00	Brand#11	LARGE ANODIZED STEEL	9.00	4.00
Brand#44	PROMO POLISHED TIN	23.00	8.00	Brand#11	LARGE BRUSHED STEEL	19.00	4.00
Brand#44	SMALL POLISHED NICKEL	14.00	8.00	Brand#11	LARGE BRUSHED TIN	3.00	4.00
Brand#44	SMALL POLISHED TIN	45.00	8.00	Brand#11	LARGE BRUSHED TIN	14.00	4.00
Brand#44	STANDARD BURNISHED COPPER	3.00	8.00	Brand#11	LARGE BURNISHED COPPER	9.00	4.00
Brand#51	LARGE ANODIZED BRASS	19.00	8.00	Brand#11	LARGE BURNISHED COPPER	19.00	4.00
Brand#51	LARGE POLISHED COPPER	23.00	8.00	Brand#11	LARGE BURNISHED STEEL	23.00	4.00
Brand#51	MEDIUM ANODIZED TIN	9.00	8.00	Brand#11	LARGE BURNISHED TIN	9.00	4.00
Brand#51	MEDIUM ANODIZED TIN	14.00	8.00	Brand#11	LARGE PLATED COPPER	23.00	4.00
Brand#51	MEDIUM BURNISHED NICKEL	23.00	8.00	Brand#11	LARGE PLATED TIN	9.00	4.00
Brand#51	SMALL ANODIZED COPPER	45.00	8.00	Brand#11	LARGE PLATED TIN	14.00	4.00
Brand#51	SMALL ANODIZED COPPER	49.00	8.00	Brand#11	LARGE PLATED TIN	23.00	4.00

Brand#11	LARGE POLISHED NICKEL	49.00	4.00	Brand#12	ECONOMY ANODIZED STEEL	9.00	4.00
Brand#11	MEDIUM ANODIZED BRASS	45.00	4.00	Brand#12	ECONOMY BRUSHED COPPER	36.00	4.00
Brand#11	MEDIUM ANODIZED TIN	14.00	4.00	Brand#12	ECONOMY BRUSHED NICKEL	49.00	4.00
Brand#11	MEDIUM BRUSHED BRASS	14.00	4.00	Brand#12	ECONOMY BRUSHED STEEL	49.00	4.00
Brand#11	MEDIUM BRUSHED BRASS	45.00	4.00	Brand#12	ECONOMY BURNISHED COPPER	45.00	4.00
Brand#11	MEDIUM BRUSHED NICKEL	14.00	4.00	Brand#12	ECONOMY PLATED COPPER	23.00	4.00
Brand#11	MEDIUM BRUSHED NICKEL	36.00	4.00	Brand#12	ECONOMY PLATED STEEL	23.00	4.00
Brand#11	MEDIUM BRUSHED STEEL	19.00	4.00	Brand#12	ECONOMY PLATED TIN	36.00	4.00
Brand#11	MEDIUM BURNISHED COPPER	9.00	4.00	Brand#12	ECONOMY POLISHED BRASS	14.00	4.00
Brand#11	MEDIUM BURNISHED TIN	36.00	4.00	Brand#12	ECONOMY POLISHED COPPER	45.00	4.00
Brand#11	MEDIUM PLATED BRASS	3.00	4.00	Brand#12	ECONOMY POLISHED NICKEL	9.00	4.00
Brand#11	MEDIUM PLATED TIN	19.00	4.00	Brand#12	LARGE ANODIZED NICKEL	9.00	4.00
Brand#11	PROMO ANODIZED BRASS	3.00	4.00	Brand#12	LARGE ANODIZED NICKEL	49.00	4.00
Brand#11	PROMO ANODIZED BRASS	19.00	4.00	Brand#12	LARGE ANODIZED STEEL	49.00	4.00
Brand#11	PROMO ANODIZED BRASS	45.00	4.00	Brand#12	LARGE ANODIZED TIN	36.00	4.00
Brand#11	PROMO ANODIZED BRASS	49.00	4.00	Brand#12	LARGE ANODIZED TIN	45.00	4.00
Brand#11	PROMO ANODIZED STEEL	23.00	4.00	Brand#12	LARGE BURNISHED BRASS	14.00	4.00
Brand#11	PROMO ANODIZED TIN	45.00	4.00	Brand#12	LARGE BURNISHED BRASS	19.00	4.00
Brand#11	PROMO BRUSHED BRASS	23.00	4.00	Brand#12	LARGE BURNISHED COPPER	9.00	4.00
Brand#11	PROMO BRUSHED STEEL	3.00	4.00	Brand#12	LARGE BURNISHED NICKEL	45.00	4.00
Brand#11	PROMO BURNISHED BRASS	23.00	4.00	Brand#12	LARGE BURNISHED TIN	36.00	4.00
Brand#11	PROMO BURNISHED BRASS	36.00	4.00	Brand#12	LARGE PLATED BRASS	3.00	4.00
Brand#11	PROMO BURNISHED BRASS	49.00	4.00	Brand#12	LARGE PLATED STEEL	36.00	4.00
Brand#11	PROMO BURNISHED TIN	9.00	4.00	Brand#12	LARGE PLATED STEEL	45.00	4.00
Brand#11	PROMO PLATED BRASS	9.00	4.00	Brand#12	LARGE PLATED TIN	23.00	4.00
Brand#11	PROMO PLATED BRASS	45.00	4.00	Brand#12	LARGE POLISHED COPPER	14.00	4.00
Brand#11	PROMO PLATED NICKEL	19.00	4.00	Brand#12	LARGE POLISHED COPPER	19.00	4.00
Brand#11	PROMO POLISHED BRASS	3.00	4.00	Brand#12	LARGE POLISHED COPPER	49.00	4.00
Brand#11	PROMO POLISHED BRASS	9.00	4.00	Brand#12	LARGE POLISHED STEEL	3.00	4.00
Brand#11	PROMO POLISHED BRASS	19.00	4.00	Brand#12	MEDIUM ANODIZED COPPER	9.00	4.00
Brand#11	PROMO POLISHED COPPER	14.00	4.00	Brand#12	MEDIUM ANODIZED COPPER	45.00	4.00
Brand#11	PROMO POLISHED COPPER	45.00	4.00	Brand#12	MEDIUM ANODIZED NICKEL	45.00	4.00
Brand#11	PROMO POLISHED TIN	49.00	4.00	Brand#12	MEDIUM BRUSHED BRASS	19.00	4.00
Brand#11	SMALL ANODIZED COPPER	36.00	4.00	Brand#12	MEDIUM BRUSHED COPPER	9.00	4.00
Brand#11	SMALL ANODIZED NICKEL	3.00	4.00	Brand#12	MEDIUM BRUSHED COPPER	36.00	4.00
Brand#11	SMALL ANODIZED NICKEL	14.00	4.00	Brand#12	MEDIUM BRUSHED COPPER	49.00	4.00
Brand#11	SMALL ANODIZED TIN	14.00	4.00	Brand#12	MEDIUM BRUSHED NICKEL	3.00	4.00
Brand#11	SMALL ANODIZED TIN	19.00	4.00	Brand#12	MEDIUM BRUSHED NICKEL	14.00	4.00
Brand#11	SMALL ANODIZED TIN	45.00	4.00	Brand#12	MEDIUM BRUSHED NICKEL	23.00	4.00
Brand#11	SMALL BRUSHED TIN	14.00	4.00	Brand#12	MEDIUM BURNISHED BRASS	3.00	4.00
Brand#11	SMALL BRUSHED TIN	23.00	4.00	Brand#12	MEDIUM BURNISHED COPPER	36.00	4.00
Brand#11	SMALL BRUSHED TIN	45.00	4.00	Brand#12	MEDIUM BURNISHED NICKEL	19.00	4.00
Brand#11	SMALL BURNISHED BRASS	49.00	4.00	Brand#12	MEDIUM BURNISHED TIN	14.00	4.00
Brand#11	SMALL BURNISHED COPPER	23.00	4.00	Brand#12	MEDIUM PLATED BRASS	23.00	4.00
Brand#11	SMALL PLATED COPPER	45.00	4.00	Brand#12	MEDIUM PLATED TIN	19.00	4.00
Brand#11	SMALL PLATED NICKEL	3.00	4.00	Brand#12	MEDIUM PLATED TIN	23.00	4.00
Brand#11	SMALL PLATED STEEL	36.00	4.00	Brand#12	PROMO ANODIZED BRASS	9.00	4.00
Brand#11	SMALL PLATED TIN	19.00	4.00	Brand#12	PROMO ANODIZED BRASS	45.00	4.00
Brand#11	SMALL POLISHED BRASS	14.00	4.00	Brand#12	PROMO ANODIZED NICKEL	14.00	4.00
Brand#11	SMALL POLISHED BRASS	23.00	4.00	Brand#12	PROMO ANODIZED STEEL	49.00	4.00
Brand#11	SMALL POLISHED COPPER	14.00	4.00	Brand#12	PROMO ANODIZED TIN	3.00	4.00
Brand#11	SMALL POLISHED COPPER	36.00	4.00	Brand#12	PROMO ANODIZED TIN	19.00	4.00
Brand#11	SMALL POLISHED STEEL	9.00	4.00	Brand#12	PROMO BRUSHED COPPER	14.00	4.00
Brand#11	STANDARD BRUSHED COPPER	23.00	4.00	Brand#12	PROMO BRUSHED COPPER	19.00	4.00
Brand#11	STANDARD BRUSHED NICKEL	14.00	4.00	Brand#12	PROMO BRUSHED NICKEL	23.00	4.00
Brand#11	STANDARD BRUSHED TIN	14.00	4.00	Brand#12	PROMO BRUSHED STEEL	23.00	4.00
Brand#11	STANDARD BURNISHED BRASS	3.00	4.00	Brand#12	PROMO BRUSHED STEEL	36.00	4.00
Brand#11	STANDARD BURNISHED STEEL	23.00	4.00	Brand#12	PROMO BURNISHED BRASS	49.00	4.00
Brand#11	STANDARD PLATED BRASS	19.00	4.00	Brand#12	PROMO BURNISHED TIN	9.00	4.00
Brand#11	STANDARD PLATED TIN	19.00	4.00	Brand#12	PROMO BURNISHED TIN	14.00	4.00
Brand#11	STANDARD POLISHED NICKEL	45.00	4.00	Brand#12	PROMO PLATED BRASS	36.00	4.00
Brand#11	STANDARD POLISHED TIN	14.00	4.00	Brand#12	PROMO POLISHED COPPER	23.00	4.00
Brand#11	STANDARD POLISHED TIN	45.00	4.00	Brand#12	PROMO POLISHED NICKEL	3.00	4.00
Brand#12	ECONOMY ANODIZED BRASS	23.00	4.00	Brand#12	PROMO POLISHED NICKEL	9.00	4.00
Brand#12	ECONOMY ANODIZED COPPER	14.00	4.00	Brand#12	PROMO POLISHED STEEL	14.00	4.00
Brand#12	ECONOMY ANODIZED NICKEL	19.00	4.00	Brand#12	PROMO POLISHED TIN	23.00	4.00
Brand#12	ECONOMY ANODIZED NICKEL	45.00	4.00	Brand#12	PROMO POLISHED TIN	36.00	4.00

Compaq

Brand#12	SMALL ANODIZED BRASS	36.00	4.00	Brand#13	LARGE BURNISHED BRASS	49.00	4.00
Brand#12	SMALL ANODIZED COPPER	23.00	4.00	Brand#13	LARGE BURNISHED TIN	49.00	4.00
Brand#12	SMALL ANODIZED STEEL	36.00	4.00	Brand#13	LARGE PLATED COPPER	23.00	4.00
Brand#12	SMALL ANODIZED TIN	14.00	4.00	Brand#13	LARGE PLATED STEEL	14.00	4.00
Brand#12	SMALL BRUSHED COPPER	19.00	4.00	Brand#13	LARGE PLATED STEEL	19.00	4.00
Brand#12	SMALL BRUSHED COPPER	36.00	4.00	Brand#13	LARGE PLATED STEEL	36.00	4.00
Brand#12	SMALL BRUSHED TIN	36.00	4.00	Brand#13	LARGE PLATED TIN	14.00	4.00
Brand#12	SMALL BURNISHED BRASS	14.00	4.00	Brand#13	LARGE PLATED TIN	45.00	4.00
Brand#12	SMALL BURNISHED COPPER	9.00	4.00	Brand#13	LARGE POLISHED BRASS	3.00	4.00
Brand#12	SMALL BURNISHED COPPER	36.00	4.00	Brand#13	LARGE POLISHED BRASS	23.00	4.00
Brand#12	SMALL PLATED BRASS	9.00	4.00	Brand#13	LARGE POLISHED BRASS	49.00	4.00
Brand#12	SMALL POLISHED BRASS	49.00	4.00	Brand#13	MEDIUM ANODIZED BRASS	3.00	4.00
Brand#12	SMALL POLISHED BRASS	19.00	4.00	Brand#13	MEDIUM ANODIZED BRASS	36.00	4.00
Brand#12	SMALL POLISHED TIN	3.00	4.00	Brand#13	MEDIUM ANODIZED COPPER	14.00	4.00
Brand#12	STANDARD ANODIZED BRASS	19.00	4.00	Brand#13	MEDIUM ANODIZED NICKEL	3.00	4.00
Brand#12	STANDARD ANODIZED NICKEL	19.00	4.00	Brand#13	MEDIUM ANODIZED STEEL	14.00	4.00
Brand#12	STANDARD ANODIZED STEEL	19.00	4.00	Brand#13	MEDIUM ANODIZED STEEL	19.00	4.00
Brand#12	STANDARD BRUSHED COPPER	36.00	4.00	Brand#13	MEDIUM ANODIZED STEEL	36.00	4.00
Brand#12	STANDARD BRUSHED NICKEL	23.00	4.00	Brand#13	MEDIUM BRUSHED BRASS	49.00	4.00
Brand#12	STANDARD BRUSHED STEEL	49.00	4.00	Brand#13	MEDIUM BRUSHED COPPER	23.00	4.00
Brand#12	STANDARD BURNISHED BRASS	23.00	4.00	Brand#13	MEDIUM BRUSHED NICKEL	45.00	4.00
Brand#12	STANDARD BURNISHED COPPER	14.00	4.00	Brand#13	MEDIUM BURNISHED BRASS	9.00	4.00
Brand#12	STANDARD BURNISHED NICKEL	45.00	4.00	Brand#13	MEDIUM BURNISHED STEEL	19.00	4.00
Brand#12	STANDARD BURNISHED NICKEL	49.00	4.00	Brand#13	MEDIUM BURNISHED STEEL	49.00	4.00
Brand#12	STANDARD BURNISHED TIN	3.00	4.00	Brand#13	MEDIUM PLATED BRASS	3.00	4.00
Brand#12	STANDARD BURNISHED TIN	14.00	4.00	Brand#13	MEDIUM PLATED BRASS	23.00	4.00
Brand#12	STANDARD PLATED BRASS	19.00	4.00	Brand#13	MEDIUM PLATED BRASS	36.00	4.00
Brand#12	STANDARD PLATED NICKEL	45.00	4.00	Brand#13	MEDIUM PLATED COPPER	19.00	4.00
Brand#12	STANDARD PLATED STEEL	36.00	4.00	Brand#13	MEDIUM PLATED COPPER	23.00	4.00
Brand#12	STANDARD PLATED STEEL	45.00	4.00	Brand#13	MEDIUM PLATED STEEL	3.00	4.00
Brand#12	STANDARD PLATED TIN	9.00	4.00	Brand#13	PROMO ANODIZED BRASS	14.00	4.00
Brand#12	STANDARD POLISHED BRASS	49.00	4.00	Brand#13	PROMO ANODIZED COPPER	9.00	4.00
Brand#12	STANDARD POLISHED COPPER	3.00	4.00	Brand#13	PROMO ANODIZED COPPER	45.00	4.00
Brand#12	STANDARD POLISHED NICKEL	23.00	4.00	Brand#13	PROMO ANODIZED STEEL	23.00	4.00
Brand#12	STANDARD POLISHED TIN	14.00	4.00	Brand#13	PROMO BRUSHED COPPER	49.00	4.00
Brand#13	ECONOMY ANODIZED NICKEL	14.00	4.00	Brand#13	PROMO BURNISHED COPPER	19.00	4.00
Brand#13	ECONOMY ANODIZED NICKEL	19.00	4.00	Brand#13	PROMO BURNISHED NICKEL	9.00	4.00
Brand#13	ECONOMY ANODIZED STEEL	45.00	4.00	Brand#13	PROMO BURNISHED NICKEL	9.00	4.00
Brand#13	ECONOMY ANODIZED STEEL	49.00	4.00	Brand#13	PROMO BURNISHED STEEL	23.00	4.00
Brand#13	ECONOMY BRUSHED BRASS	3.00	4.00	Brand#13	PROMO BURNISHED STEEL	45.00	4.00
Brand#13	ECONOMY BURNISHED STEEL	14.00	4.00	Brand#13	PROMO BURNISHED TIN	19.00	4.00
Brand#13	ECONOMY BURNISHED TIN	19.00	4.00	Brand#13	PROMO PLATED BRASS	14.00	4.00
Brand#13	ECONOMY BURNISHED TIN	45.00	4.00	Brand#13	PROMO PLATED BRASS	19.00	4.00
Brand#13	ECONOMY PLATED COPPER	19.00	4.00	Brand#13	PROMO PLATED COPPER	3.00	4.00
Brand#13	ECONOMY PLATED NICKEL	3.00	4.00	Brand#13	PROMO PLATED COPPER	19.00	4.00
Brand#13	ECONOMY PLATED STEEL	23.00	4.00	Brand#13	PROMO PLATED TIN	19.00	4.00
Brand#13	ECONOMY PLATED TIN	3.00	4.00	Brand#13	PROMO POLISHED BRASS	49.00	4.00
Brand#13	ECONOMY POLISHED BRASS	3.00	4.00	Brand#13	PROMO POLISHED STEEL	45.00	4.00
Brand#13	ECONOMY POLISHED COPPER	9.00	4.00	Brand#13	PROMO POLISHED TIN	14.00	4.00
Brand#13	ECONOMY POLISHED COPPER	49.00	4.00	Brand#13	SMALL ANODIZED STEEL	23.00	4.00
Brand#13	ECONOMY POLISHED STEEL	23.00	4.00	Brand#13	SMALL ANODIZED TIN	3.00	4.00
Brand#13	ECONOMY POLISHED STEEL	49.00	4.00	Brand#13	SMALL ANODIZED TIN	45.00	4.00
Brand#13	LARGE ANODIZED BRASS	23.00	4.00	Brand#13	SMALL BRUSHED COPPER	3.00	4.00
Brand#13	LARGE ANODIZED COPPER	19.00	4.00	Brand#13	SMALL BRUSHED NICKEL	19.00	4.00
Brand#13	LARGE ANODIZED NICKEL	9.00	4.00	Brand#13	SMALL BRUSHED TIN	9.00	4.00
Brand#13	LARGE ANODIZED STEEL	45.00	4.00	Brand#13	SMALL BRUSHED TIN	45.00	4.00
Brand#13	LARGE ANODIZED TIN	19.00	4.00	Brand#13	SMALL BURNISHED BRASS	19.00	4.00
Brand#13	LARGE BRUSHED BRASS	3.00	4.00	Brand#13	SMALL BURNISHED BRASS	45.00	4.00
Brand#13	LARGE BRUSHED BRASS	9.00	4.00	Brand#13	SMALL PLATED BRASS	9.00	4.00
Brand#13	LARGE BRUSHED BRASS	19.00	4.00	Brand#13	SMALL PLATED TIN	45.00	4.00
Brand#13	LARGE BRUSHED COPPER	9.00	4.00	Brand#13	SMALL POLISHED NICKEL	19.00	4.00
Brand#13	LARGE BRUSHED COPPER	36.00	4.00	Brand#13	SMALL POLISHED STEEL	49.00	4.00
Brand#13	LARGE BRUSHED NICKEL	3.00	4.00	Brand#13	STANDARD ANODIZED COPPER	45.00	4.00
Brand#13	LARGE BRUSHED NICKEL	9.00	4.00	Brand#13	STANDARD ANODIZED NICKEL	9.00	4.00
Brand#13	LARGE BRUSHED NICKEL	14.00	4.00	Brand#13	STANDARD ANODIZED NICKEL	19.00	4.00
Brand#13	LARGE BRUSHED NICKEL	23.00	4.00	Brand#13	STANDARD ANODIZED STEEL	14.00	4.00
Brand#13	LARGE BRUSHED STEEL	19.00	4.00	Brand#13	STANDARD ANODIZED TIN	9.00	4.00
Brand#13	LARGE BRUSHED TIN	49.00	4.00	Brand#13	STANDARD ANODIZED TIN	36.00	4.00
Brand#13				Brand#13	STANDARD BRUSHED BRASS	19.00	4.00

Brand#13	STANDARD BRUSHED TIN	9.00	4.00	Brand#14	PROMO BRUSHED STEEL	19.00	4.00
Brand#13	STANDARD BURNISHED BRASS	9.00	4.00	Brand#14	PROMO BURNISHED BRASS	23.00	4.00
Brand#13	STANDARD BURNISHED BRASS	14.00	4.00	Brand#14	PROMO BURNISHED STEEL	36.00	4.00
Brand#13	STANDARD BURNISHED COPPER	45.00	4.00	Brand#14	PROMO PLATED BRASS	9.00	4.00
Brand#13	STANDARD PLATED BRASS	49.00	4.00	Brand#14	PROMO PLATED BRASS	45.00	4.00
Brand#13	STANDARD PLATED COPPER	19.00	4.00	Brand#14	PROMO PLATED COPPER	45.00	4.00
Brand#13	STANDARD PLATED NICKEL	23.00	4.00	Brand#14	PROMO PLATED STEEL	3.00	4.00
Brand#13	STANDARD PLATED TIN	9.00	4.00	Brand#14	PROMO POLISHED BRASS	9.00	4.00
Brand#13	STANDARD POLISHED BRASS	49.00	4.00	Brand#14	PROMO POLISHED COPPER	49.00	4.00
Brand#13	STANDARD POLISHED COPPER	9.00	4.00	Brand#14	PROMO POLISHED STEEL	19.00	4.00
Brand#13	STANDARD POLISHED COPPER	49.00	4.00	Brand#14	SMALL ANODIZED STEEL	23.00	4.00
Brand#13	STANDARD POLISHED NICKEL	14.00	4.00	Brand#14	SMALL ANODIZED TIN	23.00	4.00
Brand#13	STANDARD POLISHED STEEL	23.00	4.00	Brand#14	SMALL BRUSHED BRASS	19.00	4.00
Brand#14	ECONOMY ANODIZED BRASS	19.00	4.00	Brand#14	SMALL BRUSHED BRASS	36.00	4.00
Brand#14	ECONOMY ANODIZED COPPER	9.00	4.00	Brand#14	SMALL BRUSHED COPPER	9.00	4.00
Brand#14	ECONOMY ANODIZED STEEL	19.00	4.00	Brand#14	SMALL BRUSHED TIN	36.00	4.00
Brand#14	ECONOMY ANODIZED STEEL	45.00	4.00	Brand#14	SMALL BURNISHED BRASS	45.00	4.00
Brand#14	ECONOMY BRUSHED BRASS	19.00	4.00	Brand#14	SMALL BURNISHED COPPER	9.00	4.00
Brand#14	ECONOMY BRUSHED COPPER	45.00	4.00	Brand#14	SMALL BURNISHED COPPER	14.00	4.00
Brand#14	ECONOMY BRUSHED NICKEL	14.00	4.00	Brand#14	SMALL BURNISHED COPPER	45.00	4.00
Brand#14	ECONOMY BRUSHED TIN	14.00	4.00	Brand#14	SMALL BURNISHED NICKEL	36.00	4.00
Brand#14	ECONOMY BURNISHED COPPER	9.00	4.00	Brand#14	SMALL BURNISHED STEEL	36.00	4.00
Brand#14	ECONOMY BURNISHED COPPER	19.00	4.00	Brand#14	SMALL BURNISHED TIN	23.00	4.00
Brand#14	ECONOMY BURNISHED STEEL	36.00	4.00	Brand#14	SMALL PLATED NICKEL	3.00	4.00
Brand#14	ECONOMY BURNISHED TIN	3.00	4.00	Brand#14	SMALL PLATED NICKEL	9.00	4.00
Brand#14	ECONOMY PLATED BRASS	36.00	4.00	Brand#14	SMALL PLATED STEEL	14.00	4.00
Brand#14	ECONOMY PLATED COPPER	49.00	4.00	Brand#14	SMALL POLISHED BRASS	36.00	4.00
Brand#14	ECONOMY PLATED STEEL	45.00	4.00	Brand#14	SMALL POLISHED COPPER	36.00	4.00
Brand#14	ECONOMY PLATED TIN	9.00	4.00	Brand#14	SMALL POLISHED NICKEL	9.00	4.00
Brand#14	ECONOMY POLISHED COPPER	3.00	4.00	Brand#14	SMALL POLISHED STEEL	14.00	4.00
Brand#14	ECONOMY POLISHED TIN	19.00	4.00	Brand#14	SMALL POLISHED TIN	14.00	4.00
Brand#14	LARGE ANODIZED COPPER	9.00	4.00	Brand#14	STANDARD ANODIZED BRASS	19.00	4.00
Brand#14	LARGE ANODIZED COPPER	23.00	4.00	Brand#14	STANDARD ANODIZED NICKEL	14.00	4.00
Brand#14	LARGE ANODIZED NICKEL	3.00	4.00	Brand#14	STANDARD ANODIZED STEEL	9.00	4.00
Brand#14	LARGE ANODIZED NICKEL	9.00	4.00	Brand#14	STANDARD BRUSHED COPPER	45.00	4.00
Brand#14	LARGE ANODIZED NICKEL	19.00	4.00	Brand#14	STANDARD BRUSHED NICKEL	45.00	4.00
Brand#14	LARGE ANODIZED TIN	9.00	4.00	Brand#14	STANDARD BRUSHED TIN	45.00	4.00
Brand#14	LARGE BRUSHED COPPER	14.00	4.00	Brand#14	STANDARD BURNISHED BRASS	9.00	4.00
Brand#14	LARGE BRUSHED NICKEL	45.00	4.00	Brand#14	STANDARD BURNISHED BRASS	23.00	4.00
Brand#14	LARGE PLATED BRASS	3.00	4.00	Brand#14	STANDARD BURNISHED BRASS	49.00	4.00
Brand#14	LARGE PLATED NICKEL	3.00	4.00	Brand#14	STANDARD BURNISHED NICKEL	9.00	4.00
Brand#14	LARGE PLATED NICKEL	14.00	4.00	Brand#14	STANDARD PLATED BRASS	36.00	4.00
Brand#14	LARGE PLATED NICKEL	49.00	4.00	Brand#14	STANDARD PLATED COPPER	45.00	4.00
Brand#14	LARGE PLATED TIN	49.00	4.00	Brand#14	STANDARD POLISHED NICKEL	3.00	4.00
Brand#14	LARGE POLISHED BRASS	9.00	4.00	Brand#14	STANDARD POLISHED NICKEL	9.00	4.00
Brand#14	LARGE POLISHED BRASS	14.00	4.00	Brand#14	STANDARD POLISHED TIN	19.00	4.00
Brand#14	LARGE POLISHED BRASS	36.00	4.00	Brand#15	ECONOMY ANODIZED COPPER	14.00	4.00
Brand#14	LARGE POLISHED NICKEL	3.00	4.00	Brand#15	ECONOMY ANODIZED STEEL	19.00	4.00
Brand#14	LARGE POLISHED NICKEL	14.00	4.00	Brand#15	ECONOMY ANODIZED STEEL	36.00	4.00
Brand#14	LARGE POLISHED STEEL	9.00	4.00	Brand#15	ECONOMY BRUSHED BRASS	36.00	4.00
Brand#14	LARGE POLISHED STEEL	23.00	4.00	Brand#15	ECONOMY BRUSHED COPPER	14.00	4.00
Brand#14	LARGE POLISHED STEEL	36.00	4.00	Brand#15	ECONOMY BRUSHED NICKEL	14.00	4.00
Brand#14	MEDIUM ANODIZED NICKEL	3.00	4.00	Brand#15	ECONOMY BRUSHED STEEL	3.00	4.00
Brand#14	MEDIUM ANODIZED NICKEL	49.00	4.00	Brand#15	ECONOMY BRUSHED TIN	3.00	4.00
Brand#14	MEDIUM ANODIZED STEEL	23.00	4.00	Brand#15	ECONOMY BURNISHED BRASS	14.00	4.00
Brand#14	MEDIUM ANODIZED STEEL	36.00	4.00	Brand#15	ECONOMY BURNISHED COPPER	3.00	4.00
Brand#14	MEDIUM BRUSHED BRASS	9.00	4.00	Brand#15	ECONOMY BURNISHED COPPER	23.00	4.00
Brand#14	MEDIUM BRUSHED COPPER	23.00	4.00	Brand#15	ECONOMY PLATED NICKEL	49.00	4.00
Brand#14	MEDIUM BRUSHED STEEL	14.00	4.00	Brand#15	ECONOMY PLATED STEEL	3.00	4.00
Brand#14	MEDIUM BURNISHED COPPER	14.00	4.00	Brand#15	ECONOMY PLATED STEEL	19.00	4.00
Brand#14	MEDIUM BURNISHED STEEL	3.00	4.00	Brand#15	ECONOMY PLATED STEEL	45.00	4.00
Brand#14	MEDIUM BURNISHED STEEL	49.00	4.00	Brand#15	LARGE ANODIZED BRASS	19.00	4.00
Brand#14	MEDIUM PLATED BRASS	36.00	4.00	Brand#15	LARGE ANODIZED BRASS	36.00	4.00
Brand#14	MEDIUM PLATED STEEL	49.00	4.00	Brand#15	LARGE ANODIZED BRASS	45.00	4.00
Brand#14	MEDIUM PLATED TIN	14.00	4.00	Brand#15	LARGE ANODIZED COPPER	3.00	4.00
Brand#14	PROMO ANODIZED BRASS	49.00	4.00	Brand#15	LARGE ANODIZED NICKEL	9.00	4.00
Brand#14	PROMO ANODIZED STEEL	36.00	4.00	Brand#15	LARGE ANODIZED TIN	19.00	4.00

Compaq

Brand#15	LARGE BRUSHED BRASS	9.00	4.00	Brand#15	SMALL PLATED COPPER	49.00	4.00
Brand#15	LARGE BRUSHED BRASS	19.00	4.00	Brand#15	SMALL PLATED NICKEL	36.00	4.00
Brand#15	LARGE BRUSHED COPPER	14.00	4.00	Brand#15	SMALL PLATED NICKEL	45.00	4.00
Brand#15	LARGE BRUSHED STEEL	9.00	4.00	Brand#15	SMALL PLATED STEEL	3.00	4.00
Brand#15	LARGE BRUSHED STEEL	14.00	4.00	Brand#15	SMALL PLATED TIN	9.00	4.00
Brand#15	LARGE BRUSHED STEEL	19.00	4.00	Brand#15	SMALL POLISHED COPPER	9.00	4.00
Brand#15	LARGE BRUSHED STEEL	36.00	4.00	Brand#15	SMALL POLISHED NICKEL	3.00	4.00
Brand#15	LARGE BURNISHED BRASS	14.00	4.00	Brand#15	SMALL POLISHED STEEL	19.00	4.00
Brand#15	LARGE BURNISHED BRASS	19.00	4.00	Brand#15	SMALL POLISHED STEEL	36.00	4.00
Brand#15	LARGE BURNISHED COPPER	9.00	4.00	Brand#15	SMALL POLISHED TIN	19.00	4.00
Brand#15	LARGE BURNISHED COPPER	45.00	4.00	Brand#15	SMALL POLISHED TIN	49.00	4.00
Brand#15	LARGE BURNISHED TIN	49.00	4.00	Brand#15	STANDARD ANODIZED NICKEL	19.00	4.00
Brand#15	LARGE PLATED BRASS	19.00	4.00	Brand#15	STANDARD ANODIZED NICKEL	49.00	4.00
Brand#15	LARGE PLATED COPPER	3.00	4.00	Brand#15	STANDARD ANODIZED TIN	36.00	4.00
Brand#15	LARGE PLATED COPPER	23.00	4.00	Brand#15	STANDARD BRUSHED NICKEL	3.00	4.00
Brand#15	LARGE PLATED NICKEL	36.00	4.00	Brand#15	STANDARD BURNISHED BRASS	23.00	4.00
Brand#15	MEDIUM ANODIZED BRASS	23.00	4.00	Brand#15	STANDARD BURNISHED STEEL	3.00	4.00
Brand#15	MEDIUM ANODIZED COPPER	9.00	4.00	Brand#15	STANDARD BURNISHED STEEL	45.00	4.00
Brand#15	MEDIUM ANODIZED NICKEL	3.00	4.00	Brand#15	STANDARD BURNISHED BRASS	36.00	4.00
Brand#15	MEDIUM ANODIZED TIN	19.00	4.00	Brand#15	STANDARD PLATED COPPER	14.00	4.00
Brand#15	MEDIUM BRUSHED BRASS	9.00	4.00	Brand#15	STANDARD PLATED COPPER	23.00	4.00
Brand#15	MEDIUM BRUSHED TIN	23.00	4.00	Brand#15	STANDARD PLATED NICKEL	19.00	4.00
Brand#15	MEDIUM BURNISHED COPPER	36.00	4.00	Brand#15	STANDARD PLATED TIN	45.00	4.00
Brand#15	MEDIUM BURNISHED TIN	45.00	4.00	Brand#15	STANDARD POLISHED BRASS	14.00	4.00
Brand#15	MEDIUM PLATED COPPER	9.00	4.00	Brand#15	STANDARD POLISHED COPPER	23.00	4.00
Brand#15	MEDIUM PLATED NICKEL	9.00	4.00	Brand#15	STANDARD POLISHED NICKEL	45.00	4.00
Brand#15	MEDIUM PLATED NICKEL	19.00	4.00	Brand#21	ECONOMY ANODIZED BRASS	3.00	4.00
Brand#15	MEDIUM PLATED STEEL	36.00	4.00	Brand#21	ECONOMY ANODIZED NICKEL	14.00	4.00
Brand#15	MEDIUM PLATED STEEL	49.00	4.00	Brand#21	ECONOMY ANODIZED STEEL	19.00	4.00
Brand#15	MEDIUM PLATED TIN	9.00	4.00	Brand#21	ECONOMY ANODIZED STEEL	23.00	4.00
Brand#15	MEDIUM PLATED TIN	14.00	4.00	Brand#21	ECONOMY ANODIZED STEEL	49.00	4.00
Brand#15	MEDIUM PLATED TIN	23.00	4.00	Brand#21	ECONOMY ANODIZED TIN	19.00	4.00
Brand#15	PROMO ANODIZED COPPER	23.00	4.00	Brand#21	ECONOMY BRUSHED BRASS	9.00	4.00
Brand#15	PROMO ANODIZED STEEL	14.00	4.00	Brand#21	ECONOMY BRUSHED BRASS	14.00	4.00
Brand#15	PROMO ANODIZED TIN	45.00	4.00	Brand#21	ECONOMY BRUSHED BRASS	36.00	4.00
Brand#15	PROMO BRUSHED COPPER	14.00	4.00	Brand#21	ECONOMY BRUSHED COPPER	49.00	4.00
Brand#15	PROMO BRUSHED COPPER	19.00	4.00	Brand#21	ECONOMY BRUSHED STEEL	45.00	4.00
Brand#15	PROMO BRUSHED NICKEL	19.00	4.00	Brand#21	ECONOMY BRUSHED TIN	49.00	4.00
Brand#15	PROMO BRUSHED NICKEL	23.00	4.00	Brand#21	ECONOMY BURNISHED BRASS	3.00	4.00
Brand#15	PROMO BRUSHED STEEL	14.00	4.00	Brand#21	ECONOMY BURNISHED COPPER	45.00	4.00
Brand#15	PROMO BRUSHED TIN	36.00	4.00	Brand#21	ECONOMY BURNISHED STEEL	19.00	4.00
Brand#15	PROMO BURNISHED NICKEL	9.00	4.00	Brand#21	ECONOMY BURNISHED STEEL	36.00	4.00
Brand#15	PROMO BURNISHED STEEL	45.00	4.00	Brand#21	ECONOMY PLATED BRASS	36.00	4.00
Brand#15	PROMO PLATED COPPER	3.00	4.00	Brand#21	ECONOMY PLATED COPPER	3.00	4.00
Brand#15	PROMO PLATED COPPER	36.00	4.00	Brand#21	ECONOMY PLATED COPPER	14.00	4.00
Brand#15	PROMO PLATED STEEL	3.00	4.00	Brand#21	ECONOMY PLATED NICKEL	49.00	4.00
Brand#15	PROMO PLATED TIN	49.00	4.00	Brand#21	ECONOMY POLISHED NICKEL	3.00	4.00
Brand#15	PROMO POLISHED COPPER	3.00	4.00	Brand#21	ECONOMY POLISHED NICKEL	9.00	4.00
Brand#15	PROMO POLISHED NICKEL	36.00	4.00	Brand#21	LARGE ANODIZED COPPER	3.00	4.00
Brand#15	PROMO POLISHED STEEL	36.00	4.00	Brand#21	LARGE ANODIZED COPPER	9.00	4.00
Brand#15	PROMO POLISHED TIN	49.00	4.00	Brand#21	LARGE ANODIZED STEEL	36.00	4.00
Brand#15	SMALL ANODIZED BRASS	14.00	4.00	Brand#21	LARGE ANODIZED TIN	45.00	4.00
Brand#15	SMALL ANODIZED BRASS	19.00	4.00	Brand#21	LARGE BRUSHED COPPER	45.00	4.00
Brand#15	SMALL ANODIZED COPPER	9.00	4.00	Brand#21	LARGE BRUSHED STEEL	23.00	4.00
Brand#15	SMALL ANODIZED TIN	45.00	4.00	Brand#21	LARGE BURNISHED BRASS	49.00	4.00
Brand#15	SMALL BRUSHED BRASS	3.00	4.00	Brand#21	LARGE BURNISHED COPPER	19.00	4.00
Brand#15	SMALL BRUSHED COPPER	19.00	4.00	Brand#21	LARGE BURNISHED STEEL	49.00	4.00
Brand#15	SMALL BRUSHED STEEL	23.00	4.00	Brand#21	LARGE BURNISHED TIN	49.00	4.00
Brand#15	SMALL BRUSHED TIN	45.00	4.00	Brand#21	LARGE PLATED BRASS	19.00	4.00
Brand#15	SMALL BURNISHED BRASS	19.00	4.00	Brand#21	LARGE PLATED NICKEL	23.00	4.00
Brand#15	SMALL BURNISHED COPPER	14.00	4.00	Brand#21	LARGE PLATED NICKEL	49.00	4.00
Brand#15	SMALL BURNISHED NICKEL	19.00	4.00	Brand#21	LARGE PLATED TIN	19.00	4.00
Brand#15	SMALL BURNISHED NICKEL	49.00	4.00	Brand#21	LARGE POLISHED BRASS	49.00	4.00
Brand#15	SMALL BURNISHED TIN	19.00	4.00	Brand#21	LARGE POLISHED COPPER	14.00	4.00
Brand#15	SMALL BURNISHED TIN	23.00	4.00	Brand#21	LARGE POLISHED NICKEL	3.00	4.00
Brand#15	SMALL BURNISHED TIN	36.00	4.00	Brand#21	LARGE POLISHED NICKEL	14.00	4.00
Brand#15	SMALL PLATED BRASS	3.00	4.00	Brand#21	LARGE POLISHED STEEL	14.00	4.00
Brand#15	SMALL PLATED COPPER	23.00	4.00	Brand#21	LARGE POLISHED TIN	49.00	4.00

Brand#21	MEDIUM ANODIZED COPPER	14.00	4.00	Brand#22	ECONOMY ANODIZED TIN	9.00	4.00
Brand#21	MEDIUM ANODIZED NICKEL	49.00	4.00	Brand#22	ECONOMY ANODIZED TIN	36.00	4.00
Brand#21	MEDIUM BRUSHED COPPER	3.00	4.00	Brand#22	ECONOMY BRUSHED NICKEL	36.00	4.00
Brand#21	MEDIUM BRUSHED COPPER	49.00	4.00	Brand#22	ECONOMY BRUSHED NICKEL	45.00	4.00
Brand#21	MEDIUM BRUSHED STEEL	23.00	4.00	Brand#22	ECONOMY BURNISHED BRASS	9.00	4.00
Brand#21	MEDIUM BRUSHED TIN	3.00	4.00	Brand#22	ECONOMY BURNISHED BRASS	23.00	4.00
Brand#21	MEDIUM BRUSHED TIN	14.00	4.00	Brand#22	ECONOMY BURNISHED BRASS	45.00	4.00
Brand#21	MEDIUM BURNISHED NICKEL	14.00	4.00	Brand#22	ECONOMY BURNISHED NICKEL	19.00	4.00
Brand#21	MEDIUM BURNISHED STEEL	23.00	4.00	Brand#22	ECONOMY BURNISHED NICKEL	49.00	4.00
Brand#21	MEDIUM BURNISHED TIN	3.00	4.00	Brand#22	ECONOMY BURNISHED STEEL	9.00	4.00
Brand#21	MEDIUM PLATED BRASS	3.00	4.00	Brand#22	ECONOMY BURNISHED STEEL	14.00	4.00
Brand#21	MEDIUM PLATED BRASS	19.00	4.00	Brand#22	ECONOMY BURNISHED STEEL	23.00	4.00
Brand#21	MEDIUM PLATED STEEL	36.00	4.00	Brand#22	ECONOMY PLATED BRASS	36.00	4.00
Brand#21	PROMO ANODIZED BRASS	9.00	4.00	Brand#22	ECONOMY PLATED COPPER	23.00	4.00
Brand#21	PROMO ANODIZED COPPER	14.00	4.00	Brand#22	ECONOMY PLATED TIN	3.00	4.00
Brand#21	PROMO ANODIZED NICKEL	23.00	4.00	Brand#22	ECONOMY POLISHED TIN	49.00	4.00
Brand#21	PROMO ANODIZED STEEL	3.00	4.00	Brand#22	LARGE ANODIZED BRASS	19.00	4.00
Brand#21	PROMO ANODIZED STEEL	14.00	4.00	Brand#22	LARGE ANODIZED COPPER	36.00	4.00
Brand#21	PROMO ANODIZED STEEL	36.00	4.00	Brand#22	LARGE ANODIZED STEEL	3.00	4.00
Brand#21	PROMO BRUSHED NICKEL	45.00	4.00	Brand#22	LARGE BRUSHED BRASS	23.00	4.00
Brand#21	PROMO BRUSHED STEEL	14.00	4.00	Brand#22	LARGE BRUSHED BRASS	49.00	4.00
Brand#21	PROMO BRUSHED STEEL	23.00	4.00	Brand#22	LARGE BRUSHED STEEL	49.00	4.00
Brand#21	PROMO BRUSHED STEEL	45.00	4.00	Brand#22	LARGE BURNISHED COPPER	19.00	4.00
Brand#21	PROMO BURNISHED BRASS	19.00	4.00	Brand#22	LARGE BURNISHED STEEL	23.00	4.00
Brand#21	PROMO BURNISHED COPPER	19.00	4.00	Brand#22	LARGE BURNISHED STEEL	45.00	4.00
Brand#21	PROMO BURNISHED NICKEL	9.00	4.00	Brand#22	LARGE BURNISHED TIN	45.00	4.00
Brand#21	PROMO BURNISHED TIN	19.00	4.00	Brand#22	LARGE PLATED COPPER	14.00	4.00
Brand#21	PROMO PLATED NICKEL	9.00	4.00	Brand#22	LARGE PLATED STEEL	49.00	4.00
Brand#21	PROMO PLATED NICKEL	36.00	4.00	Brand#22	LARGE POLISHED BRASS	19.00	4.00
Brand#21	PROMO PLATED STEEL	49.00	4.00	Brand#22	LARGE POLISHED COPPER	19.00	4.00
Brand#21	PROMO PLATED TIN	3.00	4.00	Brand#22	LARGE POLISHED COPPER	23.00	4.00
Brand#21	PROMO POLISHED NICKEL	23.00	4.00	Brand#22	LARGE POLISHED NICKEL	19.00	4.00
Brand#21	PROMO POLISHED TIN	14.00	4.00	Brand#22	LARGE POLISHED TIN	49.00	4.00
Brand#21	PROMO POLISHED TIN	19.00	4.00	Brand#22	MEDIUM ANODIZED BRASS	45.00	4.00
Brand#21	PROMO POLISHED TIN	23.00	4.00	Brand#22	MEDIUM ANODIZED COPPER	19.00	4.00
Brand#21	SMALL BRUSHED BRASS	23.00	4.00	Brand#22	MEDIUM ANODIZED COPPER	49.00	4.00
Brand#21	SMALL BRUSHED COPPER	49.00	4.00	Brand#22	MEDIUM ANODIZED NICKEL	9.00	4.00
Brand#21	SMALL BURNISHED BRASS	23.00	4.00	Brand#22	MEDIUM ANODIZED NICKEL	14.00	4.00
Brand#21	SMALL BURNISHED BRASS	36.00	4.00	Brand#22	MEDIUM ANODIZED NICKEL	36.00	4.00
Brand#21	SMALL BURNISHED STEEL	19.00	4.00	Brand#22	MEDIUM ANODIZED TIN	3.00	4.00
Brand#21	SMALL BURNISHED TIN	19.00	4.00	Brand#22	MEDIUM ANODIZED TIN	9.00	4.00
Brand#21	SMALL PLATED BRASS	45.00	4.00	Brand#22	MEDIUM BRUSHED BRASS	3.00	4.00
Brand#21	SMALL PLATED COPPER	45.00	4.00	Brand#22	MEDIUM BRUSHED BRASS	14.00	4.00
Brand#21	SMALL PLATED STEEL	45.00	4.00	Brand#22	MEDIUM BRUSHED COPPER	3.00	4.00
Brand#21	SMALL PLATED TIN	14.00	4.00	Brand#22	MEDIUM BRUSHED COPPER	45.00	4.00
Brand#21	SMALL PLATED TIN	45.00	4.00	Brand#22	MEDIUM BRUSHED NICKEL	14.00	4.00
Brand#21	SMALL POLISHED COPPER	9.00	4.00	Brand#22	MEDIUM BRUSHED TIN	45.00	4.00
Brand#21	SMALL POLISHED NICKEL	23.00	4.00	Brand#22	MEDIUM BURNISHED COPPER	36.00	4.00
Brand#21	SMALL POLISHED TIN	3.00	4.00	Brand#22	MEDIUM BURNISHED TIN	19.00	4.00
Brand#21	STANDARD ANODIZED BRASS	9.00	4.00	Brand#22	MEDIUM BURNISHED TIN	23.00	4.00
Brand#21	STANDARD ANODIZED NICKEL	19.00	4.00	Brand#22	MEDIUM BURNISHED TIN	49.00	4.00
Brand#21	STANDARD ANODIZED TIN	45.00	4.00	Brand#22	MEDIUM PLATED BRASS	49.00	4.00
Brand#21	STANDARD BURNISHED COPPER	36.00	4.00	Brand#22	MEDIUM PLATED COPPER	9.00	4.00
Brand#21	STANDARD BURNISHED NICKEL	23.00	4.00	Brand#22	MEDIUM PLATED STEEL	3.00	4.00
Brand#21	STANDARD BURNISHED TIN	9.00	4.00	Brand#22	PROMO ANODIZED BRASS	9.00	4.00
Brand#21	STANDARD PLATED BRASS	14.00	4.00	Brand#22	PROMO ANODIZED STEEL	36.00	4.00
Brand#21	STANDARD PLATED COPPER	19.00	4.00	Brand#22	PROMO ANODIZED TIN	45.00	4.00
Brand#21	STANDARD PLATED NICKEL	3.00	4.00	Brand#22	PROMO BRUSHED BRASS	3.00	4.00
Brand#21	STANDARD PLATED STEEL	9.00	4.00	Brand#22	PROMO BRUSHED BRASS	9.00	4.00
Brand#21	STANDARD PLATED TIN	9.00	4.00	Brand#22	PROMO BRUSHED BRASS	36.00	4.00
Brand#21	STANDARD POLISHED BRASS	9.00	4.00	Brand#22	PROMO BRUSHED STEEL	36.00	4.00
Brand#21	STANDARD POLISHED COPPER	49.00	4.00	Brand#22	PROMO BURNISHED BRASS	23.00	4.00
Brand#21	STANDARD POLISHED STEEL	36.00	4.00	Brand#22	PROMO BURNISHED COPPER	9.00	4.00
Brand#21	STANDARD POLISHED TIN	36.00	4.00	Brand#22	PROMO PLATED BRASS	14.00	4.00
Brand#22	ECONOMY ANODIZED STEEL	9.00	4.00	Brand#22	PROMO PLATED BRASS	45.00	4.00
Brand#22	ECONOMY ANODIZED STEEL	14.00	4.00	Brand#22	PROMO PLATED NICKEL	3.00	4.00
Brand#22	ECONOMY ANODIZED STEEL	23.00	4.00	Brand#22	PROMO PLATED STEEL	19.00	4.00

Compaq

Brand#22	PROMO POLISHED BRASS	3.00	4.00	Brand#23	MEDIUM ANODIZED TIN	36.00	4.00
Brand#22	PROMO POLISHED STEEL	14.00	4.00	Brand#23	MEDIUM ANODIZED TIN	45.00	4.00
Brand#22	PROMO POLISHED STEEL	23.00	4.00	Brand#23	MEDIUM BRUSHED COPPER	3.00	4.00
Brand#22	SMALL ANODIZED TIN	36.00	4.00	Brand#23	MEDIUM BRUSHED COPPER	23.00	4.00
Brand#22	SMALL ANODIZED TIN	49.00	4.00	Brand#23	MEDIUM BRUSHED NICKEL	3.00	4.00
Brand#22	SMALL BRUSHED NICKEL	3.00	4.00	Brand#23	MEDIUM BRUSHED TIN	14.00	4.00
Brand#22	SMALL BRUSHED NICKEL	36.00	4.00	Brand#23	MEDIUM BURNISHED BRASS	9.00	4.00
Brand#22	SMALL BRUSHED NICKEL	45.00	4.00	Brand#23	MEDIUM BURNISHED BRASS	45.00	4.00
Brand#22	SMALL BRUSHED TIN	45.00	4.00	Brand#23	MEDIUM BURNISHED COPPER	19.00	4.00
Brand#22	SMALL BURNISHED STEEL	23.00	4.00	Brand#23	MEDIUM PLATED COPPER	19.00	4.00
Brand#22	SMALL BURNISHED TIN	14.00	4.00	Brand#23	MEDIUM PLATED COPPER	36.00	4.00
Brand#22	SMALL PLATED STEEL	3.00	4.00	Brand#23	MEDIUM PLATED COPPER	45.00	4.00
Brand#22	SMALL PLATED TIN	9.00	4.00	Brand#23	MEDIUM PLATED NICKEL	9.00	4.00
Brand#22	SMALL PLATED TIN	36.00	4.00	Brand#23	MEDIUM PLATED NICKEL	14.00	4.00
Brand#22	SMALL POLISHED BRASS	23.00	4.00	Brand#23	PROMO ANODIZED COPPER	9.00	4.00
Brand#22	SMALL POLISHED NICKEL	19.00	4.00	Brand#23	PROMO ANODIZED COPPER	19.00	4.00
Brand#22	STANDARD ANODIZED BRASS	14.00	4.00	Brand#23	PROMO ANODIZED STEEL	36.00	4.00
Brand#22	STANDARD ANODIZED BRASS	23.00	4.00	Brand#23	PROMO ANODIZED TIN	14.00	4.00
Brand#22	STANDARD BRUSHED COPPER	49.00	4.00	Brand#23	PROMO BRUSHED BRASS	3.00	4.00
Brand#22	STANDARD BRUSHED NICKEL	3.00	4.00	Brand#23	PROMO BRUSHED BRASS	19.00	4.00
Brand#22	STANDARD BRUSHED NICKEL	23.00	4.00	Brand#23	PROMO BRUSHED BRASS	36.00	4.00
Brand#22	STANDARD BRUSHED STEEL	9.00	4.00	Brand#23	PROMO BRUSHED COPPER	3.00	4.00
Brand#22	STANDARD BRUSHED TIN	19.00	4.00	Brand#23	PROMO BRUSHED TIN	49.00	4.00
Brand#22	STANDARD BURNISHED NICKEL	3.00	4.00	Brand#23	PROMO BURNISHED BRASS	14.00	4.00
Brand#22	STANDARD BURNISHED NICKEL	14.00	4.00	Brand#23	PROMO BURNISHED BRASS	45.00	4.00
Brand#22	STANDARD BURNISHED NICKEL	45.00	4.00	Brand#23	PROMO BURNISHED COPPER	14.00	4.00
Brand#22	STANDARD BURNISHED STEEL	3.00	4.00	Brand#23	PROMO PLATED BRASS	23.00	4.00
Brand#22	STANDARD BURNISHED STEEL	36.00	4.00	Brand#23	PROMO POLISHED BRASS	14.00	4.00
Brand#22	STANDARD BURNISHED STEEL	45.00	4.00	Brand#23	PROMO POLISHED BRASS	23.00	4.00
Brand#22	STANDARD BURNISHED STEEL	49.00	4.00	Brand#23	PROMO POLISHED COPPER	36.00	4.00
Brand#22	STANDARD PLATED BRASS	45.00	4.00	Brand#23	PROMO POLISHED STEEL	36.00	4.00
Brand#22	STANDARD PLATED NICKEL	3.00	4.00	Brand#23	SMALL ANODIZED STEEL	23.00	4.00
Brand#22	STANDARD PLATED NICKEL	45.00	4.00	Brand#23	SMALL BRUSHED BRASS	49.00	4.00
Brand#22	STANDARD PLATED STEEL	14.00	4.00	Brand#23	SMALL BRUSHED COPPER	45.00	4.00
Brand#22	STANDARD PLATED TIN	19.00	4.00	Brand#23	SMALL BRUSHED STEEL	3.00	4.00
Brand#22	STANDARD PLATED TIN	49.00	4.00	Brand#23	SMALL BRUSHED STEEL	19.00	4.00
Brand#22	STANDARD POLISHED COPPER	9.00	4.00	Brand#23	SMALL BURNISHED BRASS	36.00	4.00
Brand#22	STANDARD POLISHED STEEL	49.00	4.00	Brand#23	SMALL BURNISHED COPPER	45.00	4.00
Brand#22	STANDARD POLISHED TIN	45.00	4.00	Brand#23	SMALL BURNISHED COPPER	49.00	4.00
Brand#23	ECONOMY ANODIZED NICKEL	49.00	4.00	Brand#23	SMALL BURNISHED STEEL	45.00	4.00
Brand#23	ECONOMY ANODIZED STEEL	14.00	4.00	Brand#23	SMALL PLATED BRASS	36.00	4.00
Brand#23	ECONOMY ANODIZED STEEL	49.00	4.00	Brand#23	SMALL PLATED BRASS	49.00	4.00
Brand#23	ECONOMY ANODIZED TIN	49.00	4.00	Brand#23	SMALL PLATED COPPER	14.00	4.00
Brand#23	ECONOMY BRUSHED BRASS	3.00	4.00	Brand#23	SMALL PLATED TIN	14.00	4.00
Brand#23	ECONOMY BRUSHED COPPER	9.00	4.00	Brand#23	SMALL POLISHED BRASS	9.00	4.00
Brand#23	ECONOMY BRUSHED TIN	9.00	4.00	Brand#23	SMALL POLISHED BRASS	14.00	4.00
Brand#23	ECONOMY BURNISHED STEEL	49.00	4.00	Brand#23	SMALL POLISHED NICKEL	3.00	4.00
Brand#23	ECONOMY PLATED COPPER	14.00	4.00	Brand#23	SMALL POLISHED STEEL	14.00	4.00
Brand#23	ECONOMY PLATED NICKEL	23.00	4.00	Brand#23	SMALL POLISHED TIN	9.00	4.00
Brand#23	ECONOMY PLATED STEEL	14.00	4.00	Brand#23	STANDARD ANODIZED BRASS	19.00	4.00
Brand#23	ECONOMY POLISHED NICKEL	9.00	4.00	Brand#23	STANDARD ANODIZED BRASS	45.00	4.00
Brand#23	LARGE ANODIZED BRASS	14.00	4.00	Brand#23	STANDARD ANODIZED COPPER	19.00	4.00
Brand#23	LARGE ANODIZED COPPER	9.00	4.00	Brand#23	STANDARD ANODIZED TIN	3.00	4.00
Brand#23	LARGE ANODIZED COPPER	14.00	4.00	Brand#23	STANDARD BRUSHED COPPER	36.00	4.00
Brand#23	LARGE ANODIZED COPPER	45.00	4.00	Brand#23	STANDARD BRUSHED NICKEL	19.00	4.00
Brand#23	LARGE ANODIZED STEEL	19.00	4.00	Brand#23	STANDARD BRUSHED STEEL	49.00	4.00
Brand#23	LARGE ANODIZED STEEL	36.00	4.00	Brand#23	STANDARD BURNISHED COPPER	19.00	4.00
Brand#23	LARGE ANODIZED STEEL	49.00	4.00	Brand#23	STANDARD PLATED BRASS	3.00	4.00
Brand#23	LARGE ANODIZED TIN	9.00	4.00	Brand#23	STANDARD PLATED BRASS	9.00	4.00
Brand#23	LARGE PLATED BRASS	9.00	4.00	Brand#23	STANDARD PLATED STEEL	36.00	4.00
Brand#23	LARGE PLATED BRASS	49.00	4.00	Brand#23	STANDARD PLATED TIN	19.00	4.00
Brand#23	LARGE PLATED COPPER	3.00	4.00	Brand#23	STANDARD POLISHED BRASS	9.00	4.00
Brand#23	LARGE POLISHED BRASS	45.00	4.00	Brand#23	STANDARD POLISHED BRASS	49.00	4.00
Brand#23	LARGE POLISHED STEEL	9.00	4.00	Brand#23	STANDARD POLISHED STEEL	19.00	4.00
Brand#23	MEDIUM ANODIZED BRASS	19.00	4.00	Brand#23	STANDARD POLISHED STEEL	49.00	4.00
Brand#23	MEDIUM ANODIZED NICKEL	3.00	4.00	Brand#23	STANDARD POLISHED TIN	23.00	4.00
Brand#23	MEDIUM ANODIZED NICKEL	14.00	4.00	Brand#24	ECONOMY ANODIZED BRASS	3.00	4.00
Brand#23	MEDIUM ANODIZED STEEL	45.00	4.00	Brand#24	ECONOMY ANODIZED BRASS	9.00	4.00

Brand#24	ECONOMY ANODIZED BRASS	23.00	4.00	Brand#24	PROMO PLATED COPPER	49.00	4.00
Brand#24	ECONOMY ANODIZED COPPER	9.00	4.00	Brand#24	PROMO POLISHED BRASS	3.00	4.00
Brand#24	ECONOMY ANODIZED COPPER	49.00	4.00	Brand#24	PROMO POLISHED BRASS	14.00	4.00
Brand#24	ECONOMY BRUSHED BRASS	36.00	4.00	Brand#24	PROMO POLISHED NICKEL	3.00	4.00
Brand#24	ECONOMY BRUSHED COPPER	23.00	4.00	Brand#24	PROMO POLISHED STEEL	14.00	4.00
Brand#24	ECONOMY BURNISHED COPPER	3.00	4.00	Brand#24	PROMO POLISHED STEEL	19.00	4.00
Brand#24	ECONOMY BURNISHED NICKEL	19.00	4.00	Brand#24	PROMO POLISHED STEEL	23.00	4.00
Brand#24	ECONOMY BURNISHED STEEL	45.00	4.00	Brand#24	SMALL ANODIZED BRASS	19.00	4.00
Brand#24	ECONOMY PLATED BRASS	23.00	4.00	Brand#24	SMALL ANODIZED COPPER	3.00	4.00
Brand#24	ECONOMY PLATED COPPER	36.00	4.00	Brand#24	SMALL ANODIZED NICKEL	14.00	4.00
Brand#24	ECONOMY PLATED STEEL	45.00	4.00	Brand#24	SMALL ANODIZED STEEL	36.00	4.00
Brand#24	ECONOMY POLISHED BRASS	23.00	4.00	Brand#24	SMALL ANODIZED TIN	3.00	4.00
Brand#24	ECONOMY POLISHED COPPER	45.00	4.00	Brand#24	SMALL ANODIZED TIN	36.00	4.00
Brand#24	ECONOMY POLISHED NICKEL	36.00	4.00	Brand#24	SMALL BRUSHED COPPER	49.00	4.00
Brand#24	ECONOMY POLISHED STEEL	14.00	4.00	Brand#24	SMALL BRUSHED NICKEL	49.00	4.00
Brand#24	ECONOMY POLISHED STEEL	36.00	4.00	Brand#24	SMALL BURNISHED BRASS	14.00	4.00
Brand#24	LARGE ANODIZED NICKEL	23.00	4.00	Brand#24	SMALL BURNISHED BRASS	19.00	4.00
Brand#24	LARGE ANODIZED NICKEL	45.00	4.00	Brand#24	SMALL BURNISHED TIN	9.00	4.00
Brand#24	LARGE ANODIZED TIN	45.00	4.00	Brand#24	SMALL PLATED BRASS	3.00	4.00
Brand#24	LARGE BRUSHED BRASS	14.00	4.00	Brand#24	SMALL PLATED COPPER	14.00	4.00
Brand#24	LARGE BRUSHED BRASS	23.00	4.00	Brand#24	SMALL PLATED COPPER	36.00	4.00
Brand#24	LARGE BRUSHED STEEL	9.00	4.00	Brand#24	SMALL PLATED NICKEL	14.00	4.00
Brand#24	LARGE BRUSHED STEEL	23.00	4.00	Brand#24	SMALL PLATED NICKEL	49.00	4.00
Brand#24	LARGE BRUSHED STEEL	45.00	4.00	Brand#24	SMALL POLISHED BRASS	3.00	4.00
Brand#24	LARGE BRUSHED TIN	49.00	4.00	Brand#24	SMALL POLISHED NICKEL	9.00	4.00
Brand#24	LARGE BURNISHED BRASS	3.00	4.00	Brand#24	SMALL POLISHED NICKEL	19.00	4.00
Brand#24	LARGE BURNISHED NICKEL	19.00	4.00	Brand#24	SMALL POLISHED NICKEL	36.00	4.00
Brand#24	LARGE PLATED BRASS	9.00	4.00	Brand#24	SMALL POLISHED STEEL	9.00	4.00
Brand#24	LARGE PLATED NICKEL	36.00	4.00	Brand#24	SMALL POLISHED STEEL	36.00	4.00
Brand#24	LARGE PLATED NICKEL	49.00	4.00	Brand#24	STANDARD ANODIZED TIN	9.00	4.00
Brand#24	LARGE PLATED TIN	9.00	4.00	Brand#24	STANDARD ANODIZED TIN	49.00	4.00
Brand#24	LARGE PLATED TIN	19.00	4.00	Brand#24	STANDARD BRUSHED BRASS	14.00	4.00
Brand#24	LARGE PLATED TIN	36.00	4.00	Brand#24	STANDARD BRUSHED COPPER	23.00	4.00
Brand#24	LARGE PLATED TIN	49.00	4.00	Brand#24	STANDARD BRUSHED NICKEL	19.00	4.00
Brand#24	LARGE POLISHED BRASS	9.00	4.00	Brand#24	STANDARD BRUSHED STEEL	14.00	4.00
Brand#24	LARGE POLISHED COPPER	9.00	4.00	Brand#24	STANDARD BRUSHED TIN	36.00	4.00
Brand#24	LARGE POLISHED COPPER	49.00	4.00	Brand#24	STANDARD BURNISHED COPPER	19.00	4.00
Brand#24	LARGE POLISHED NICKEL	19.00	4.00	Brand#24	STANDARD BURNISHED COPPER	36.00	4.00
Brand#24	LARGE POLISHED STEEL	23.00	4.00	Brand#24	STANDARD BURNISHED NICKEL	45.00	4.00
Brand#24	LARGE POLISHED TIN	14.00	4.00	Brand#24	STANDARD PLATED BRASS	36.00	4.00
Brand#24	MEDIUM ANODIZED COPPER	45.00	4.00	Brand#24	STANDARD PLATED COPPER	45.00	4.00
Brand#24	MEDIUM BRUSHED COPPER	9.00	4.00	Brand#24	STANDARD PLATED NICKEL	36.00	4.00
Brand#24	MEDIUM BRUSHED COPPER	14.00	4.00	Brand#24	STANDARD PLATED TIN	36.00	4.00
Brand#24	MEDIUM BRUSHED NICKEL	9.00	4.00	Brand#24	STANDARD POLISHED COPPER	45.00	4.00
Brand#24	MEDIUM BRUSHED NICKEL	23.00	4.00	Brand#24	STANDARD POLISHED NICKEL	14.00	4.00
Brand#24	MEDIUM BRUSHED STEEL	14.00	4.00	Brand#25	ECONOMY ANODIZED BRASS	14.00	4.00
Brand#24	MEDIUM BRUSHED STEEL	45.00	4.00	Brand#25	ECONOMY ANODIZED BRASS	49.00	4.00
Brand#24	MEDIUM BRUSHED STEEL	49.00	4.00	Brand#25	ECONOMY ANODIZED TIN	9.00	4.00
Brand#24	MEDIUM BURNISHED BRASS	36.00	4.00	Brand#25	ECONOMY ANODIZED TIN	19.00	4.00
Brand#24	MEDIUM BURNISHED NICKEL	36.00	4.00	Brand#25	ECONOMY ANODIZED TIN	49.00	4.00
Brand#24	MEDIUM BURNISHED STEEL	36.00	4.00	Brand#25	ECONOMY BRUSHED COPPER	36.00	4.00
Brand#24	MEDIUM PLATED COPPER	14.00	4.00	Brand#25	ECONOMY BURNISHED COPPER	45.00	4.00
Brand#24	MEDIUM PLATED STEEL	3.00	4.00	Brand#25	ECONOMY BURNISHED TIN	19.00	4.00
Brand#24	MEDIUM PLATED STEEL	19.00	4.00	Brand#25	ECONOMY PLATED NICKEL	23.00	4.00
Brand#24	PROMO ANODIZED NICKEL	9.00	4.00	Brand#25	ECONOMY PLATED TIN	14.00	4.00
Brand#24	PROMO ANODIZED NICKEL	19.00	4.00	Brand#25	ECONOMY POLISHED BRASS	23.00	4.00
Brand#24	PROMO ANODIZED NICKEL	45.00	4.00	Brand#25	ECONOMY POLISHED COPPER	9.00	4.00
Brand#24	PROMO ANODIZED STEEL	3.00	4.00	Brand#25	ECONOMY POLISHED NICKEL	3.00	4.00
Brand#24	PROMO ANODIZED TIN	45.00	4.00	Brand#25	ECONOMY POLISHED TIN	9.00	4.00
Brand#24	PROMO BRUSHED BRASS	19.00	4.00	Brand#25	ECONOMY POLISHED TIN	45.00	4.00
Brand#24	PROMO BRUSHED NICKEL	19.00	4.00	Brand#25	LARGE ANODIZED BRASS	3.00	4.00
Brand#24	PROMO BRUSHED NICKEL	45.00	4.00	Brand#25	LARGE ANODIZED BRASS	14.00	4.00
Brand#24	PROMO BURNISHED BRASS	3.00	4.00	Brand#25	LARGE ANODIZED COPPER	36.00	4.00
Brand#24	PROMO BURNISHED BRASS	45.00	4.00	Brand#25	LARGE ANODIZED NICKEL	23.00	4.00
Brand#24	PROMO BURNISHED STEEL	49.00	4.00	Brand#25	LARGE ANODIZED STEEL	23.00	4.00
Brand#24	PROMO PLATED BRASS	3.00	4.00	Brand#25	LARGE BRUSHED NICKEL	19.00	4.00
Brand#24	PROMO PLATED COPPER	23.00	4.00	Brand#25	LARGE BRUSHED NICKEL	49.00	4.00

Brand#25	LARGE BRUSHED TIN	3.00	4.00	Brand#25	SMALL POLISHED BRASS	23.00	4.00
Brand#25	LARGE BRUSHED TIN	9.00	4.00	Brand#25	SMALL POLISHED COPPER	9.00	4.00
Brand#25	LARGE BURNISHED BRASS	19.00	4.00	Brand#25	SMALL POLISHED STEEL	14.00	4.00
Brand#25	LARGE BURNISHED BRASS	23.00	4.00	Brand#25	STANDARD ANODIZED STEEL	3.00	4.00
Brand#25	LARGE BURNISHED BRASS	49.00	4.00	Brand#25	STANDARD ANODIZED STEEL	19.00	4.00
Brand#25	LARGE BURNISHED NICKEL	14.00	4.00	Brand#25	STANDARD ANODIZED TIN	9.00	4.00
Brand#25	LARGE BURNISHED TIN	49.00	4.00	Brand#25	STANDARD BRUSHED BRASS	14.00	4.00
Brand#25	LARGE PLATED BRASS	14.00	4.00	Brand#25	STANDARD BRUSHED NICKEL	19.00	4.00
Brand#25	LARGE PLATED NICKEL	23.00	4.00	Brand#25	STANDARD BRUSHED TIN	9.00	4.00
Brand#25	LARGE PLATED NICKEL	45.00	4.00	Brand#25	STANDARD BURNISHED NICKEL	9.00	4.00
Brand#25	LARGE PLATED TIN	19.00	4.00	Brand#25	STANDARD PLATED BRASS	3.00	4.00
Brand#25	LARGE PLATED TIN	23.00	4.00	Brand#25	STANDARD PLATED COPPER	14.00	4.00
Brand#25	LARGE POLISHED BRASS	9.00	4.00	Brand#25	STANDARD PLATED NICKEL	36.00	4.00
Brand#25	LARGE POLISHED COPPER	14.00	4.00	Brand#25	STANDARD POLISHED BRASS	45.00	4.00
Brand#25	LARGE POLISHED COPPER	36.00	4.00	Brand#25	STANDARD POLISHED COPPER	23.00	4.00
Brand#25	MEDIUM ANODIZED TIN	36.00	4.00	Brand#25	STANDARD POLISHED NICKEL	3.00	4.00
Brand#25	MEDIUM BRUSHED COPPER	9.00	4.00	Brand#25	STANDARD POLISHED NICKEL	49.00	4.00
Brand#25	MEDIUM BRUSHED COPPER	36.00	4.00	Brand#25	STANDARD POLISHED TIN	36.00	4.00
Brand#25	MEDIUM BRUSHED COPPER	49.00	4.00	Brand#25	STANDARD POLISHED TIN	45.00	4.00
Brand#25	MEDIUM BURNISHED COPPER	49.00	4.00	Brand#31	ECONOMY ANODIZED BRASS	3.00	4.00
Brand#25	MEDIUM BURNISHED NICKEL	9.00	4.00	Brand#31	ECONOMY ANODIZED COPPER	45.00	4.00
Brand#25	MEDIUM BURNISHED NICKEL	49.00	4.00	Brand#31	ECONOMY ANODIZED STEEL	3.00	4.00
Brand#25	MEDIUM BURNISHED STEEL	3.00	4.00	Brand#31	ECONOMY ANODIZED TIN	45.00	4.00
Brand#25	MEDIUM BURNISHED STEEL	36.00	4.00	Brand#31	ECONOMY BRUSHED BRASS	14.00	4.00
Brand#25	MEDIUM BURNISHED STEEL	45.00	4.00	Brand#31	ECONOMY BRUSHED COPPER	19.00	4.00
Brand#25	MEDIUM BURNISHED STEEL	49.00	4.00	Brand#31	ECONOMY BRUSHED NICKEL	9.00	4.00
Brand#25	MEDIUM BURNISHED TIN	9.00	4.00	Brand#31	ECONOMY BRUSHED NICKEL	14.00	4.00
Brand#25	MEDIUM BURNISHED TIN	36.00	4.00	Brand#31	ECONOMY BRUSHED NICKEL	49.00	4.00
Brand#25	MEDIUM PLATED BRASS	45.00	4.00	Brand#31	ECONOMY BURNISHED COPPER	36.00	4.00
Brand#25	MEDIUM PLATED COPPER	14.00	4.00	Brand#31	ECONOMY BURNISHED STEEL	3.00	4.00
Brand#25	MEDIUM PLATED NICKEL	45.00	4.00	Brand#31	ECONOMY BURNISHED TIN	49.00	4.00
Brand#25	MEDIUM PLATED STEEL	9.00	4.00	Brand#31	ECONOMY PLATED COPPER	49.00	4.00
Brand#25	MEDIUM PLATED STEEL	36.00	4.00	Brand#31	ECONOMY PLATED NICKEL	9.00	4.00
Brand#25	PROMO ANODIZED COPPER	14.00	4.00	Brand#31	ECONOMY PLATED STEEL	23.00	4.00
Brand#25	PROMO ANODIZED COPPER	19.00	4.00	Brand#31	ECONOMY PLATED TIN	36.00	4.00
Brand#25	PROMO ANODIZED STEEL	36.00	4.00	Brand#31	ECONOMY PLATED TIN	49.00	4.00
Brand#25	PROMO ANODIZED TIN	3.00	4.00	Brand#31	ECONOMY POLISHED COPPER	3.00	4.00
Brand#25	PROMO ANODIZED TIN	14.00	4.00	Brand#31	ECONOMY POLISHED COPPER	36.00	4.00
Brand#25	PROMO BRUSHED NICKEL	3.00	4.00	Brand#31	ECONOMY POLISHED COPPER	49.00	4.00
Brand#25	PROMO BRUSHED STEEL	19.00	4.00	Brand#31	ECONOMY POLISHED NICKEL	3.00	4.00
Brand#25	PROMO BRUSHED TIN	14.00	4.00	Brand#31	LARGE ANODIZED BRASS	19.00	4.00
Brand#25	PROMO BRUSHED TIN	36.00	4.00	Brand#31	LARGE ANODIZED STEEL	45.00	4.00
Brand#25	PROMO BURNISHED COPPER	19.00	4.00	Brand#31	LARGE BRUSHED BRASS	36.00	4.00
Brand#25	PROMO BURNISHED COPPER	45.00	4.00	Brand#31	LARGE BRUSHED BRASS	49.00	4.00
Brand#25	PROMO BURNISHED COPPER	49.00	4.00	Brand#31	LARGE BRUSHED TIN	3.00	4.00
Brand#25	PROMO BURNISHED NICKEL	36.00	4.00	Brand#31	LARGE BURNISHED BRASS	9.00	4.00
Brand#25	PROMO BURNISHED TIN	3.00	4.00	Brand#31	LARGE PLATED COPPER	19.00	4.00
Brand#25	PROMO PLATED BRASS	45.00	4.00	Brand#31	LARGE PLATED NICKEL	14.00	4.00
Brand#25	PROMO PLATED COPPER	19.00	4.00	Brand#31	LARGE PLATED TIN	9.00	4.00
Brand#25	PROMO PLATED NICKEL	45.00	4.00	Brand#31	LARGE PLATED TIN	14.00	4.00
Brand#25	PROMO PLATED NICKEL	49.00	4.00	Brand#31	LARGE POLISHED BRASS	14.00	4.00
Brand#25	PROMO PLATED STEEL	23.00	4.00	Brand#31	LARGE POLISHED STEEL	14.00	4.00
Brand#25	PROMO POLISHED BRASS	23.00	4.00	Brand#31	LARGE POLISHED STEEL	45.00	4.00
Brand#25	SMALL ANODIZED BRASS	45.00	4.00	Brand#31	LARGE POLISHED TIN	19.00	4.00
Brand#25	SMALL ANODIZED NICKEL	19.00	4.00	Brand#31	MEDIUM ANODIZED BRASS	23.00	4.00
Brand#25	SMALL ANODIZED STEEL	23.00	4.00	Brand#31	MEDIUM ANODIZED BRASS	36.00	4.00
Brand#25	SMALL ANODIZED TIN	14.00	4.00	Brand#31	MEDIUM ANODIZED COPPER	14.00	4.00
Brand#25	SMALL ANODIZED TIN	19.00	4.00	Brand#31	MEDIUM ANODIZED COPPER	19.00	4.00
Brand#25	SMALL BRUSHED COPPER	45.00	4.00	Brand#31	MEDIUM ANODIZED COPPER	36.00	4.00
Brand#25	SMALL BRUSHED NICKEL	9.00	4.00	Brand#31	MEDIUM ANODIZED STEEL	14.00	4.00
Brand#25	SMALL BURNISHED COPPER	3.00	4.00	Brand#31	MEDIUM ANODIZED STEEL	49.00	4.00
Brand#25	SMALL BURNISHED STEEL	3.00	4.00	Brand#31	MEDIUM ANODIZED TIN	19.00	4.00
Brand#25	SMALL BURNISHED STEEL	14.00	4.00	Brand#31	MEDIUM ANODIZED TIN	49.00	4.00
Brand#25	SMALL BURNISHED TIN	3.00	4.00	Brand#31	MEDIUM BRUSHED BRASS	36.00	4.00
Brand#25	SMALL PLATED BRASS	19.00	4.00	Brand#31	MEDIUM BRUSHED STEEL	14.00	4.00
Brand#25	SMALL PLATED COPPER	23.00	4.00	Brand#31	MEDIUM BURNISHED BRASS	14.00	4.00
Brand#25	SMALL PLATED STEEL	45.00	4.00	Brand#31	MEDIUM BURNISHED COPPER	3.00	4.00
Brand#25	SMALL PLATED TIN	36.00	4.00	Brand#31	MEDIUM BURNISHED NICKEL	9.00	4.00

Brand#31	MEDIUM BURNISHED STEEL	9.00	4.00	Brand#32	LARGE ANODIZED BRASS	36.00	4.00
Brand#31	MEDIUM BURNISHED TIN	14.00	4.00	Brand#32	LARGE ANODIZED NICKEL	45.00	4.00
Brand#31	MEDIUM BURNISHED TIN	23.00	4.00	Brand#32	LARGE ANODIZED STEEL	3.00	4.00
Brand#31	MEDIUM PLATED BRASS	3.00	4.00	Brand#32	LARGE ANODIZED STEEL	14.00	4.00
Brand#31	MEDIUM PLATED TIN	9.00	4.00	Brand#32	LARGE BRUSHED TIN	45.00	4.00
Brand#31	MEDIUM PLATED TIN	36.00	4.00	Brand#32	LARGE BURNISHED NICKEL	36.00	4.00
Brand#31	MEDIUM PLATED TIN	45.00	4.00	Brand#32	LARGE BURNISHED TIN	19.00	4.00
Brand#31	PROMO ANODIZED BRASS	3.00	4.00	Brand#32	LARGE BURNISHED TIN	45.00	4.00
Brand#31	PROMO ANODIZED NICKEL	9.00	4.00	Brand#32	LARGE PLATED BRASS	3.00	4.00
Brand#31	PROMO BRUSHED BRASS	3.00	4.00	Brand#32	LARGE PLATED NICKEL	49.00	4.00
Brand#31	PROMO BRUSHED BRASS	23.00	4.00	Brand#32	LARGE PLATED STEEL	19.00	4.00
Brand#31	PROMO BRUSHED COPPER	23.00	4.00	Brand#32	LARGE PLATED STEEL	36.00	4.00
Brand#31	PROMO BRUSHED NICKEL	45.00	4.00	Brand#32	LARGE POLISHED BRASS	45.00	4.00
Brand#31	PROMO BURNISHED COPPER	36.00	4.00	Brand#32	LARGE POLISHED COPPER	9.00	4.00
Brand#31	PROMO BURNISHED STEEL	3.00	4.00	Brand#32	LARGE POLISHED COPPER	49.00	4.00
Brand#31	PROMO BURNISHED TIN	3.00	4.00	Brand#32	LARGE POLISHED NICKEL	3.00	4.00
Brand#31	PROMO PLATED BRASS	19.00	4.00	Brand#32	MEDIUM ANODIZED BRASS	3.00	4.00
Brand#31	PROMO PLATED NICKEL	36.00	4.00	Brand#32	MEDIUM ANODIZED BRASS	9.00	4.00
Brand#31	PROMO POLISHED BRASS	49.00	4.00	Brand#32	MEDIUM ANODIZED TIN	23.00	4.00
Brand#31	PROMO POLISHED COPPER	14.00	4.00	Brand#32	MEDIUM BRUSHED BRASS	23.00	4.00
Brand#31	PROMO POLISHED NICKEL	3.00	4.00	Brand#32	MEDIUM BRUSHED BRASS	49.00	4.00
Brand#31	PROMO POLISHED NICKEL	9.00	4.00	Brand#32	MEDIUM BRUSHED COPPER	9.00	4.00
Brand#31	PROMO POLISHED TIN	3.00	4.00	Brand#32	MEDIUM BRUSHED COPPER	19.00	4.00
Brand#31	PROMO POLISHED TIN	23.00	4.00	Brand#32	MEDIUM BRUSHED TIN	49.00	4.00
Brand#31	SMALL ANODIZED COPPER	45.00	4.00	Brand#32	MEDIUM BURNISHED BRASS	9.00	4.00
Brand#31	SMALL ANODIZED STEEL	23.00	4.00	Brand#32	MEDIUM BURNISHED BRASS	36.00	4.00
Brand#31	SMALL ANODIZED TIN	3.00	4.00	Brand#32	MEDIUM BURNISHED BRASS	49.00	4.00
Brand#31	SMALL BRUSHED COPPER	36.00	4.00	Brand#32	MEDIUM BURNISHED COPPER	9.00	4.00
Brand#31	SMALL BRUSHED COPPER	49.00	4.00	Brand#32	MEDIUM BURNISHED COPPER	45.00	4.00
Brand#31	SMALL BRUSHED NICKEL	19.00	4.00	Brand#32	MEDIUM BURNISHED NICKEL	49.00	4.00
Brand#31	SMALL BRUSHED NICKEL	23.00	4.00	Brand#32	MEDIUM BURNISHED TIN	9.00	4.00
Brand#31	SMALL BURNISHED BRASS	45.00	4.00	Brand#32	MEDIUM BURNISHED TIN	45.00	4.00
Brand#31	SMALL BURNISHED NICKEL	9.00	4.00	Brand#32	MEDIUM PLATED BRASS	3.00	4.00
Brand#31	SMALL BURNISHED NICKEL	36.00	4.00	Brand#32	MEDIUM PLATED BRASS	49.00	4.00
Brand#31	SMALL PLATED COPPER	36.00	4.00	Brand#32	MEDIUM PLATED COPPER	3.00	4.00
Brand#31	SMALL PLATED NICKEL	9.00	4.00	Brand#32	MEDIUM PLATED STEEL	9.00	4.00
Brand#31	SMALL POLISHED BRASS	3.00	4.00	Brand#32	MEDIUM PLATED TIN	9.00	4.00
Brand#31	SMALL POLISHED COPPER	45.00	4.00	Brand#32	PROMO ANODIZED BRASS	3.00	4.00
Brand#31	SMALL POLISHED NICKEL	45.00	4.00	Brand#32	PROMO ANODIZED COPPER	19.00	4.00
Brand#31	SMALL POLISHED TIN	23.00	4.00	Brand#32	PROMO ANODIZED NICKEL	23.00	4.00
Brand#31	SMALL POLISHED TIN	49.00	4.00	Brand#32	PROMO BRUSHED COPPER	23.00	4.00
Brand#31	STANDARD BRUSHED STEEL	23.00	4.00	Brand#32	PROMO BRUSHED NICKEL	14.00	4.00
Brand#31	STANDARD BRUSHED STEEL	49.00	4.00	Brand#32	PROMO BRUSHED NICKEL	36.00	4.00
Brand#31	STANDARD BURNISHED BRASS	14.00	4.00	Brand#32	PROMO BRUSHED STEEL	14.00	4.00
Brand#31	STANDARD BURNISHED NICKEL	45.00	4.00	Brand#32	PROMO BRUSHED STEEL	23.00	4.00
Brand#31	STANDARD PLATED NICKEL	3.00	4.00	Brand#32	PROMO BRUSHED STEEL	49.00	4.00
Brand#31	STANDARD POLISHED BRASS	3.00	4.00	Brand#32	PROMO BURNISHED BRASS	45.00	4.00
Brand#31	STANDARD POLISHED BRASS	45.00	4.00	Brand#32	PROMO BURNISHED NICKEL	45.00	4.00
Brand#31	STANDARD POLISHED STEEL	36.00	4.00	Brand#32	PROMO BURNISHED TIN	14.00	4.00
Brand#32	ECONOMY ANODIZED BRASS	19.00	4.00	Brand#32	PROMO BURNISHED TIN	45.00	4.00
Brand#32	ECONOMY ANODIZED COPPER	36.00	4.00	Brand#32	PROMO PLATED TIN	19.00	4.00
Brand#32	ECONOMY ANODIZED STEEL	23.00	4.00	Brand#32	PROMO POLISHED NICKEL	36.00	4.00
Brand#32	ECONOMY ANODIZED STEEL	36.00	4.00	Brand#32	PROMO POLISHED TIN	3.00	4.00
Brand#32	ECONOMY ANODIZED STEEL	45.00	4.00	Brand#32	SMALL ANODIZED BRASS	3.00	4.00
Brand#32	ECONOMY ANODIZED TIN	19.00	4.00	Brand#32	SMALL ANODIZED NICKEL	3.00	4.00
Brand#32	ECONOMY BRUSHED COPPER	45.00	4.00	Brand#32	SMALL ANODIZED NICKEL	14.00	4.00
Brand#32	ECONOMY BRUSHED TIN	45.00	4.00	Brand#32	SMALL ANODIZED TIN	9.00	4.00
Brand#32	ECONOMY BURNISHED BRASS	23.00	4.00	Brand#32	SMALL BRUSHED BRASS	9.00	4.00
Brand#32	ECONOMY BURNISHED COPPER	36.00	4.00	Brand#32	SMALL BRUSHED BRASS	19.00	4.00
Brand#32	ECONOMY BURNISHED COPPER	45.00	4.00	Brand#32	SMALL BRUSHED COPPER	3.00	4.00
Brand#32	ECONOMY BURNISHED STEEL	19.00	4.00	Brand#32	SMALL BRUSHED COPPER	23.00	4.00
Brand#32	ECONOMY PLATED BRASS	9.00	4.00	Brand#32	SMALL BRUSHED NICKEL	9.00	4.00
Brand#32	ECONOMY PLATED COPPER	9.00	4.00	Brand#32	SMALL BRUSHED NICKEL	45.00	4.00
Brand#32	ECONOMY PLATED NICKEL	23.00	4.00	Brand#32	SMALL BRUSHED STEEL	23.00	4.00
Brand#32	ECONOMY PLATED TIN	45.00	4.00	Brand#32	SMALL BRUSHED TIN	9.00	4.00
Brand#32	ECONOMY POLISHED STEEL	3.00	4.00	Brand#32	SMALL BURNISHED NICKEL	36.00	4.00
Brand#32	LARGE ANODIZED BRASS	23.00	4.00	Brand#32	SMALL BURNISHED STEEL	3.00	4.00

Brand#32	SMALL BURNISHED TIN	23.00	4.00	Brand#33	LARGE BRUSHED TIN	36.00	4.00
Brand#32	SMALL PLATED BRASS	49.00	4.00	Brand#33	LARGE BURNISHED COPPER	45.00	4.00
Brand#32	SMALL PLATED COPPER	36.00	4.00	Brand#33	LARGE BURNISHED NICKEL	23.00	4.00
Brand#32	SMALL PLATED COPPER	45.00	4.00	Brand#33	LARGE BURNISHED STEEL	19.00	4.00
Brand#32	SMALL PLATED NICKEL	45.00	4.00	Brand#33	LARGE PLATED BRASS	3.00	4.00
Brand#32	SMALL PLATED STEEL	45.00	4.00	Brand#33	LARGE PLATED COPPER	19.00	4.00
Brand#32	SMALL PLATED TIN	23.00	4.00	Brand#33	LARGE PLATED STEEL	3.00	4.00
Brand#32	SMALL PLATED TIN	36.00	4.00	Brand#33	LARGE PLATED STEEL	19.00	4.00
Brand#32	SMALL PLATED TIN	45.00	4.00	Brand#33	LARGE PLATED TIN	45.00	4.00
Brand#32	SMALL POLISHED NICKEL	36.00	4.00	Brand#33	LARGE POLISHED BRASS	45.00	4.00
Brand#32	SMALL POLISHED STEEL	14.00	4.00	Brand#33	LARGE POLISHED STEEL	14.00	4.00
Brand#32	SMALL POLISHED STEEL	23.00	4.00	Brand#33	LARGE POLISHED STEEL	23.00	4.00
Brand#32	SMALL POLISHED STEEL	36.00	4.00	Brand#33	LARGE POLISHED TIN	23.00	4.00
Brand#32	SMALL POLISHED TIN	36.00	4.00	Brand#33	MEDIUM ANODIZED BRASS	3.00	4.00
Brand#32	SMALL POLISHED TIN	45.00	4.00	Brand#33	MEDIUM ANODIZED COPPER	9.00	4.00
Brand#32	STANDARD ANODIZED NICKEL	19.00	4.00	Brand#33	MEDIUM ANODIZED COPPER	36.00	4.00
Brand#32	STANDARD ANODIZED TIN	9.00	4.00	Brand#33	MEDIUM ANODIZED COPPER	49.00	4.00
Brand#32	STANDARD ANODIZED TIN	14.00	4.00	Brand#33	MEDIUM ANODIZED NICKEL	3.00	4.00
Brand#32	STANDARD ANODIZED TIN	19.00	4.00	Brand#33	MEDIUM ANODIZED NICKEL	19.00	4.00
Brand#32	STANDARD BRUSHED NICKEL	23.00	4.00	Brand#33	MEDIUM BRUSHED BRASS	3.00	4.00
Brand#32	STANDARD BURNISHED BRASS	36.00	4.00	Brand#33	MEDIUM BRUSHED STEEL	19.00	4.00
Brand#32	STANDARD BURNISHED BRASS	45.00	4.00	Brand#33	MEDIUM BRUSHED TIN	14.00	4.00
Brand#32	STANDARD BURNISHED COPPER	3.00	4.00	Brand#33	MEDIUM BURNISHED COPPER	14.00	4.00
Brand#32	STANDARD BURNISHED COPPER	36.00	4.00	Brand#33	MEDIUM BURNISHED COPPER	49.00	4.00
Brand#32	STANDARD BURNISHED NICKEL	49.00	4.00	Brand#33	MEDIUM BURNISHED TIN	36.00	4.00
Brand#32	STANDARD BURNISHED STEEL	49.00	4.00	Brand#33	MEDIUM PLATED BRASS	3.00	4.00
Brand#32	STANDARD BURNISHED TIN	23.00	4.00	Brand#33	MEDIUM PLATED STEEL	3.00	4.00
Brand#32	STANDARD PLATED BRASS	9.00	4.00	Brand#33	MEDIUM PLATED STEEL	49.00	4.00
Brand#32	STANDARD PLATED BRASS	45.00	4.00	Brand#33	PROMO ANODIZED BRASS	3.00	4.00
Brand#32	STANDARD PLATED STEEL	36.00	4.00	Brand#33	PROMO BRUSHED BRASS	49.00	4.00
Brand#32	STANDARD POLISHED BRASS	14.00	4.00	Brand#33	PROMO BURNISHED COPPER	23.00	4.00
Brand#32	STANDARD POLISHED COPPER	36.00	4.00	Brand#33	PROMO BURNISHED NICKEL	14.00	4.00
Brand#32	STANDARD POLISHED STEEL	14.00	4.00	Brand#33	PROMO BURNISHED NICKEL	36.00	4.00
Brand#33	ECONOMY ANODIZED BRASS	23.00	4.00	Brand#33	PROMO BURNISHED TIN	19.00	4.00
Brand#33	ECONOMY ANODIZED COPPER	9.00	4.00	Brand#33	PROMO BURNISHED TIN	23.00	4.00
Brand#33	ECONOMY ANODIZED NICKEL	3.00	4.00	Brand#33	PROMO PLATED COPPER	14.00	4.00
Brand#33	ECONOMY ANODIZED NICKEL	9.00	4.00	Brand#33	PROMO PLATED STEEL	45.00	4.00
Brand#33	ECONOMY ANODIZED NICKEL	23.00	4.00	Brand#33	PROMO PLATED STEEL	49.00	4.00
Brand#33	ECONOMY ANODIZED NICKEL	36.00	4.00	Brand#33	PROMO PLATED TIN	49.00	4.00
Brand#33	ECONOMY BRUSHED BRASS	14.00	4.00	Brand#33	PROMO POLISHED COPPER	3.00	4.00
Brand#33	ECONOMY BRUSHED COPPER	23.00	4.00	Brand#33	PROMO POLISHED STEEL	3.00	4.00
Brand#33	ECONOMY BURNISHED BRASS	49.00	4.00	Brand#33	PROMO POLISHED STEEL	9.00	4.00
Brand#33	ECONOMY BURNISHED COPPER	3.00	4.00	Brand#33	PROMO POLISHED STEEL	23.00	4.00
Brand#33	ECONOMY BURNISHED COPPER	14.00	4.00	Brand#33	SMALL ANODIZED BRASS	19.00	4.00
Brand#33	ECONOMY BURNISHED STEEL	3.00	4.00	Brand#33	SMALL ANODIZED COPPER	23.00	4.00
Brand#33	ECONOMY BURNISHED TIN	36.00	4.00	Brand#33	SMALL ANODIZED COPPER	49.00	4.00
Brand#33	ECONOMY BURNISHED TIN	45.00	4.00	Brand#33	SMALL ANODIZED STEEL	9.00	4.00
Brand#33	ECONOMY PLATED COPPER	19.00	4.00	Brand#33	SMALL BRUSHED BRASS	3.00	4.00
Brand#33	ECONOMY PLATED COPPER	45.00	4.00	Brand#33	SMALL BRUSHED COPPER	3.00	4.00
Brand#33	ECONOMY PLATED NICKEL	14.00	4.00	Brand#33	SMALL BRUSHED NICKEL	45.00	4.00
Brand#33	ECONOMY PLATED NICKEL	36.00	4.00	Brand#33	SMALL BRUSHED STEEL	3.00	4.00
Brand#33	ECONOMY PLATED STEEL	3.00	4.00	Brand#33	SMALL BRUSHED STEEL	9.00	4.00
Brand#33	ECONOMY PLATED STEEL	23.00	4.00	Brand#33	SMALL BRUSHED TIN	9.00	4.00
Brand#33	ECONOMY PLATED STEEL	36.00	4.00	Brand#33	SMALL BURNISHED BRASS	19.00	4.00
Brand#33	ECONOMY POLISHED BRASS	14.00	4.00	Brand#33	SMALL BURNISHED NICKEL	3.00	4.00
Brand#33	ECONOMY POLISHED NICKEL	19.00	4.00	Brand#33	SMALL PLATED BRASS	3.00	4.00
Brand#33	ECONOMY POLISHED TIN	9.00	4.00	Brand#33	SMALL PLATED STEEL	14.00	4.00
Brand#33	LARGE ANODIZED BRASS	36.00	4.00	Brand#33	SMALL PLATED STEEL	45.00	4.00
Brand#33	LARGE ANODIZED COPPER	19.00	4.00	Brand#33	SMALL PLATED TIN	23.00	4.00
Brand#33	LARGE ANODIZED COPPER	45.00	4.00	Brand#33	SMALL PLATED TIN	36.00	4.00
Brand#33	LARGE ANODIZED NICKEL	36.00	4.00	Brand#33	SMALL POLISHED NICKEL	23.00	4.00
Brand#33	LARGE ANODIZED NICKEL	45.00	4.00	Brand#33	SMALL POLISHED TIN	19.00	4.00
Brand#33	LARGE ANODIZED NICKEL	45.00	4.00	Brand#33	SMALL POLISHED TIN	23.00	4.00
Brand#33	LARGE ANODIZED STEEL	3.00	4.00	Brand#33	SMALL POLISHED TIN	45.00	4.00
Brand#33	LARGE ANODIZED STEEL	45.00	4.00	Brand#33	STANDARD ANODIZED COPPER	49.00	4.00
Brand#33	LARGE ANODIZED TIN	45.00	4.00	Brand#33	STANDARD ANODIZED STEEL	14.00	4.00
Brand#33	LARGE BRUSHED BRASS	3.00	4.00	Brand#33	STANDARD ANODIZED STEEL	45.00	4.00
Brand#33	LARGE BRUSHED BRASS	49.00	4.00	Brand#33	STANDARD ANODIZED STEEL	49.00	4.00
Brand#33	LARGE BRUSHED STEEL	19.00	4.00	Brand#33	STANDARD ANODIZED TIN	45.00	4.00

Brand#33	STANDARD BRUSHED BRASS	9.00	4.00	Brand#34	MEDIUM BRUSHED TIN	14.00	4.00
Brand#33	STANDARD BRUSHED NICKEL	45.00	4.00	Brand#34	MEDIUM BURNISHED STEEL	45.00	4.00
Brand#33	STANDARD BRUSHED STEEL	9.00	4.00	Brand#34	MEDIUM BURNISHED STEEL	49.00	4.00
Brand#33	STANDARD BRUSHED TIN	36.00	4.00	Brand#34	MEDIUM PLATED COPPER	36.00	4.00
Brand#33	STANDARD BURNISHED BRASS	9.00	4.00	Brand#34	MEDIUM PLATED TIN	3.00	4.00
Brand#33	STANDARD BURNISHED BRASS	23.00	4.00	Brand#34	MEDIUM PLATED TIN	14.00	4.00
Brand#33	STANDARD BURNISHED NICKEL	49.00	4.00	Brand#34	PROMO ANODIZED COPPER	45.00	4.00
Brand#33	STANDARD PLATED BRASS	49.00	4.00	Brand#34	PROMO ANODIZED NICKEL	14.00	4.00
Brand#33	STANDARD PLATED COPPER	3.00	4.00	Brand#34	PROMO ANODIZED STEEL	49.00	4.00
Brand#33	STANDARD PLATED COPPER	14.00	4.00	Brand#34	PROMO ANODIZED TIN	14.00	4.00
Brand#33	STANDARD PLATED NICKEL	36.00	4.00	Brand#34	PROMO BRUSHED BRASS	9.00	4.00
Brand#33	STANDARD PLATED STEEL	3.00	4.00	Brand#34	PROMO BRUSHED BRASS	23.00	4.00
Brand#33	STANDARD PLATED STEEL	36.00	4.00	Brand#34	PROMO BRUSHED COPPER	36.00	4.00
Brand#33	STANDARD PLATED TIN	14.00	4.00	Brand#34	PROMO BRUSHED STEEL	36.00	4.00
Brand#33	STANDARD POLISHED BRASS	9.00	4.00	Brand#34	PROMO BURNISHED BRASS	49.00	4.00
Brand#33	STANDARD POLISHED BRASS	19.00	4.00	Brand#34	PROMO BURNISHED STEEL	3.00	4.00
Brand#33	STANDARD POLISHED STEEL	3.00	4.00	Brand#34	PROMO PLATED BRASS	9.00	4.00
Brand#33	STANDARD POLISHED STEEL	9.00	4.00	Brand#34	PROMO PLATED STEEL	49.00	4.00
Brand#33	STANDARD POLISHED STEEL	14.00	4.00	Brand#34	PROMO POLISHED BRASS	23.00	4.00
Brand#34	ECONOMY ANODIZED BRASS	9.00	4.00	Brand#34	PROMO POLISHED NICKEL	3.00	4.00
Brand#34	ECONOMY ANODIZED COPPER	3.00	4.00	Brand#34	PROMO POLISHED NICKEL	36.00	4.00
Brand#34	ECONOMY ANODIZED COPPER	14.00	4.00	Brand#34	SMALL ANODIZED BRASS	36.00	4.00
Brand#34	ECONOMY ANODIZED COPPER	19.00	4.00	Brand#34	SMALL ANODIZED COPPER	45.00	4.00
Brand#34	ECONOMY ANODIZED STEEL	9.00	4.00	Brand#34	SMALL ANODIZED NICKEL	14.00	4.00
Brand#34	ECONOMY ANODIZED TIN	49.00	4.00	Brand#34	SMALL ANODIZED NICKEL	36.00	4.00
Brand#34	ECONOMY BRUSHED BRASS	14.00	4.00	Brand#34	SMALL ANODIZED STEEL	3.00	4.00
Brand#34	ECONOMY BRUSHED NICKEL	49.00	4.00	Brand#34	SMALL ANODIZED STEEL	19.00	4.00
Brand#34	ECONOMY BURNISHED COPPER	9.00	4.00	Brand#34	SMALL ANODIZED STEEL	23.00	4.00
Brand#34	ECONOMY BURNISHED STEEL	19.00	4.00	Brand#34	SMALL ANODIZED STEEL	36.00	4.00
Brand#34	ECONOMY BURNISHED TIN	3.00	4.00	Brand#34	SMALL BRUSHED BRASS	14.00	4.00
Brand#34	ECONOMY BURNISHED TIN	23.00	4.00	Brand#34	SMALL BRUSHED BRASS	36.00	4.00
Brand#34	ECONOMY PLATED BRASS	9.00	4.00	Brand#34	SMALL BRUSHED NICKEL	14.00	4.00
Brand#34	ECONOMY PLATED BRASS	14.00	4.00	Brand#34	SMALL BRUSHED NICKEL	36.00	4.00
Brand#34	ECONOMY PLATED COPPER	3.00	4.00	Brand#34	SMALL BRUSHED NICKEL	45.00	4.00
Brand#34	ECONOMY PLATED NICKEL	45.00	4.00	Brand#34	SMALL BRUSHED TIN	9.00	4.00
Brand#34	ECONOMY PLATED TIN	14.00	4.00	Brand#34	SMALL BRUSHED TIN	23.00	4.00
Brand#34	ECONOMY PLATED TIN	45.00	4.00	Brand#34	SMALL BRUSHED TIN	36.00	4.00
Brand#34	ECONOMY POLISHED BRASS	45.00	4.00	Brand#34	SMALL BURNISHED COPPER	9.00	4.00
Brand#34	LARGE ANODIZED BRASS	14.00	4.00	Brand#34	SMALL BURNISHED TIN	36.00	4.00
Brand#34	LARGE ANODIZED BRASS	23.00	4.00	Brand#34	SMALL PLATED BRASS	14.00	4.00
Brand#34	LARGE ANODIZED BRASS	36.00	4.00	Brand#34	SMALL PLATED COPPER	36.00	4.00
Brand#34	LARGE ANODIZED NICKEL	3.00	4.00	Brand#34	SMALL PLATED TIN	45.00	4.00
Brand#34	LARGE ANODIZED TIN	49.00	4.00	Brand#34	SMALL POLISHED NICKEL	14.00	4.00
Brand#34	LARGE BRUSHED BRASS	49.00	4.00	Brand#34	SMALL POLISHED NICKEL	45.00	4.00
Brand#34	LARGE BRUSHED COPPER	23.00	4.00	Brand#34	SMALL POLISHED TIN	9.00	4.00
Brand#34	LARGE BRUSHED NICKEL	23.00	4.00	Brand#34	SMALL POLISHED TIN	14.00	4.00
Brand#34	LARGE BRUSHED STEEL	14.00	4.00	Brand#34	SMALL POLISHED TIN	19.00	4.00
Brand#34	LARGE BRUSHED STEEL	19.00	4.00	Brand#34	STANDARD ANODIZED BRASS	23.00	4.00
Brand#34	LARGE BRUSHED TIN	9.00	4.00	Brand#34	STANDARD ANODIZED BRASS	36.00	4.00
Brand#34	LARGE BURNISHED BRASS	23.00	4.00	Brand#34	STANDARD ANODIZED COPPER	45.00	4.00
Brand#34	LARGE BURNISHED COPPER	3.00	4.00	Brand#34	STANDARD ANODIZED NICKEL	36.00	4.00
Brand#34	LARGE BURNISHED COPPER	36.00	4.00	Brand#34	STANDARD ANODIZED STEEL	9.00	4.00
Brand#34	LARGE BURNISHED NICKEL	19.00	4.00	Brand#34	STANDARD ANODIZED STEEL	49.00	4.00
Brand#34	LARGE PLATED BRASS	23.00	4.00	Brand#34	STANDARD ANODIZED TIN	9.00	4.00
Brand#34	LARGE PLATED BRASS	45.00	4.00	Brand#34	STANDARD BRUSHED BRASS	19.00	4.00
Brand#34	LARGE PLATED COPPER	23.00	4.00	Brand#34	STANDARD BRUSHED BRASS	23.00	4.00
Brand#34	LARGE PLATED COPPER	49.00	4.00	Brand#34	STANDARD BRUSHED NICKEL	23.00	4.00
Brand#34	LARGE PLATED STEEL	49.00	4.00	Brand#34	STANDARD BRUSHED STEEL	3.00	4.00
Brand#34	LARGE POLISHED NICKEL	49.00	4.00	Brand#34	STANDARD BRUSHED TIN	19.00	4.00
Brand#34	MEDIUM ANODIZED COPPER	36.00	4.00	Brand#34	STANDARD BURNISHED COPPER	45.00	4.00
Brand#34	MEDIUM ANODIZED TIN	3.00	4.00	Brand#34	STANDARD BURNISHED NICKEL	19.00	4.00
Brand#34	MEDIUM BRUSHED BRASS	49.00	4.00	Brand#34	STANDARD BURNISHED NICKEL	45.00	4.00
Brand#34	MEDIUM BRUSHED COPPER	9.00	4.00	Brand#34	STANDARD BURNISHED STEEL	36.00	4.00
Brand#34	MEDIUM BRUSHED NICKEL	9.00	4.00	Brand#34	STANDARD BURNISHED TIN	45.00	4.00
Brand#34	MEDIUM BRUSHED NICKEL	23.00	4.00	Brand#34	STANDARD PLATED BRASS	9.00	4.00
Brand#34	MEDIUM BRUSHED TIN	3.00	4.00	Brand#34	STANDARD PLATED COPPER	9.00	4.00
Brand#34				Brand#34	STANDARD PLATED NICKEL	36.00	4.00

Brand#35	ECONOMY ANODIZED COPPER	3.00	4.00	Brand#35	SMALL ANODIZED COPPER	9.00	4.00
Brand#35	ECONOMY ANODIZED STEEL	45.00	4.00	Brand#35	SMALL ANODIZED STEEL	19.00	4.00
Brand#35	ECONOMY BRUSHED BRASS	3.00	4.00	Brand#35	SMALL ANODIZED TIN	19.00	4.00
Brand#35	ECONOMY BRUSHED NICKEL	49.00	4.00	Brand#35	SMALL BRUSHED BRASS	36.00	4.00
Brand#35	ECONOMY BRUSHED STEEL	23.00	4.00	Brand#35	SMALL BRUSHED STEEL	49.00	4.00
Brand#35	ECONOMY BRUSHED STEEL	45.00	4.00	Brand#35	SMALL BRUSHED TIN	3.00	4.00
Brand#35	ECONOMY BRUSHED TIN	14.00	4.00	Brand#35	SMALL BRUSHED TIN	19.00	4.00
Brand#35	ECONOMY BRUSHED TIN	23.00	4.00	Brand#35	SMALL BRUSHED TIN	23.00	4.00
Brand#35	ECONOMY BURNISHED NICKEL	19.00	4.00	Brand#35	SMALL BURNISHED BRASS	23.00	4.00
Brand#35	ECONOMY BURNISHED STEEL	36.00	4.00	Brand#35	SMALL BURNISHED STEEL	36.00	4.00
Brand#35	ECONOMY BURNISHED TIN	9.00	4.00	Brand#35	SMALL BURNISHED TIN	3.00	4.00
Brand#35	ECONOMY BURNISHED TIN	19.00	4.00	Brand#35	SMALL BURNISHED TIN	36.00	4.00
Brand#35	ECONOMY BURNISHED TIN	49.00	4.00	Brand#35	SMALL BURNISHED TIN	49.00	4.00
Brand#35	ECONOMY POLISHED COPPER	9.00	4.00	Brand#35	SMALL PLATED BRASS	23.00	4.00
Brand#35	ECONOMY POLISHED TIN	19.00	4.00	Brand#35	SMALL PLATED STEEL	14.00	4.00
Brand#35	LARGE ANODIZED BRASS	3.00	4.00	Brand#35	SMALL POLISHED BRASS	36.00	4.00
Brand#35	LARGE ANODIZED BRASS	23.00	4.00	Brand#35	SMALL POLISHED STEEL	3.00	4.00
Brand#35	LARGE ANODIZED COPPER	49.00	4.00	Brand#35	SMALL POLISHED STEEL	49.00	4.00
Brand#35	LARGE ANODIZED STEEL	36.00	4.00	Brand#35	SMALL POLISHED TIN	23.00	4.00
Brand#35	LARGE ANODIZED TIN	9.00	4.00	Brand#35	SMALL POLISHED TIN	45.00	4.00
Brand#35	LARGE BRUSHED COPPER	9.00	4.00	Brand#35	STANDARD ANODIZED NICKEL	14.00	4.00
Brand#35	LARGE BRUSHED COPPER	23.00	4.00	Brand#35	STANDARD ANODIZED STEEL	23.00	4.00
Brand#35	LARGE BRUSHED STEEL	3.00	4.00	Brand#35	STANDARD ANODIZED STEEL	45.00	4.00
Brand#35	LARGE BRUSHED STEEL	9.00	4.00	Brand#35	STANDARD ANODIZED TIN	9.00	4.00
Brand#35	LARGE BURNISHED BRASS	36.00	4.00	Brand#35	STANDARD ANODIZED TIN	19.00	4.00
Brand#35	LARGE BURNISHED BRASS	45.00	4.00	Brand#35	STANDARD BRUSHED BRASS	3.00	4.00
Brand#35	LARGE BURNISHED COPPER	23.00	4.00	Brand#35	STANDARD BRUSHED BRASS	23.00	4.00
Brand#35	LARGE BURNISHED NICKEL	23.00	4.00	Brand#35	STANDARD BRUSHED BRASS	36.00	4.00
Brand#35	LARGE PLATED BRASS	9.00	4.00	Brand#35	STANDARD BRUSHED COPPER	36.00	4.00
Brand#35	LARGE PLATED COPPER	36.00	4.00	Brand#35	STANDARD BRUSHED NICKEL	36.00	4.00
Brand#35	LARGE POLISHED BRASS	49.00	4.00	Brand#35	STANDARD BRUSHED NICKEL	49.00	4.00
Brand#35	LARGE POLISHED STEEL	9.00	4.00	Brand#35	STANDARD BRUSHED TIN	9.00	4.00
Brand#35	LARGE POLISHED TIN	14.00	4.00	Brand#35	STANDARD BURNISHED BRASS	9.00	4.00
Brand#35	MEDIUM ANODIZED BRASS	9.00	4.00	Brand#35	STANDARD BURNISHED BRASS	19.00	4.00
Brand#35	MEDIUM ANODIZED BRASS	36.00	4.00	Brand#35	STANDARD BURNISHED BRASS	23.00	4.00
Brand#35	MEDIUM ANODIZED COPPER	9.00	4.00	Brand#35	STANDARD BURNISHED COPPER	36.00	4.00
Brand#35	MEDIUM BRUSHED BRASS	14.00	4.00	Brand#35	STANDARD BURNISHED STEEL	14.00	4.00
Brand#35	MEDIUM BRUSHED COPPER	9.00	4.00	Brand#35	STANDARD PLATED COPPER	19.00	4.00
Brand#35	MEDIUM BRUSHED COPPER	36.00	4.00	Brand#35	STANDARD PLATED NICKEL	23.00	4.00
Brand#35	MEDIUM BURNISHED BRASS	49.00	4.00	Brand#35	STANDARD PLATED STEEL	14.00	4.00
Brand#35	MEDIUM BURNISHED NICKEL	45.00	4.00	Brand#35	STANDARD PLATED STEEL	23.00	4.00
Brand#35	MEDIUM BURNISHED TIN	36.00	4.00	Brand#35	STANDARD PLATED TIN	49.00	4.00
Brand#35	MEDIUM PLATED BRASS	23.00	4.00	Brand#35	STANDARD POLISHED NICKEL	23.00	4.00
Brand#35	MEDIUM PLATED COPPER	9.00	4.00	Brand#35	STANDARD POLISHED TIN	23.00	4.00
Brand#35	MEDIUM PLATED NICKEL	45.00	4.00	Brand#35	STANDARD POLISHED TIN	45.00	4.00
Brand#35	MEDIUM PLATED NICKEL	49.00	4.00	Brand#41	ECONOMY ANODIZED STEEL	49.00	4.00
Brand#35	MEDIUM PLATED STEEL	49.00	4.00	Brand#41	ECONOMY BRUSHED BRASS	3.00	4.00
Brand#35	PROMO ANODIZED COPPER	49.00	4.00	Brand#41	ECONOMY BRUSHED COPPER	36.00	4.00
Brand#35	PROMO ANODIZED NICKEL	19.00	4.00	Brand#41	ECONOMY BRUSHED NICKEL	23.00	4.00
Brand#35	PROMO ANODIZED NICKEL	23.00	4.00	Brand#41	ECONOMY BRUSHED STEEL	36.00	4.00
Brand#35	PROMO ANODIZED TIN	3.00	4.00	Brand#41	ECONOMY BRUSHED STEEL	45.00	4.00
Brand#35	PROMO ANODIZED TIN	14.00	4.00	Brand#41	ECONOMY BRUSHED TIN	14.00	4.00
Brand#35	PROMO BRUSHED BRASS	49.00	4.00	Brand#41	ECONOMY PLATED COPPER	3.00	4.00
Brand#35	PROMO BRUSHED NICKEL	14.00	4.00	Brand#41	ECONOMY PLATED STEEL	3.00	4.00
Brand#35	PROMO BRUSHED NICKEL	19.00	4.00	Brand#41	ECONOMY PLATED TIN	23.00	4.00
Brand#35	PROMO BURNISHED BRASS	3.00	4.00	Brand#41	ECONOMY POLISHED COPPER	19.00	4.00
Brand#35	PROMO BURNISHED STEEL	3.00	4.00	Brand#41	ECONOMY POLISHED NICKEL	9.00	4.00
Brand#35	PROMO PLATED BRASS	19.00	4.00	Brand#41	ECONOMY POLISHED NICKEL	14.00	4.00
Brand#35	PROMO PLATED COPPER	14.00	4.00	Brand#41	ECONOMY POLISHED NICKEL	23.00	4.00
Brand#35	PROMO PLATED STEEL	23.00	4.00	Brand#41	ECONOMY POLISHED NICKEL	49.00	4.00
Brand#35	PROMO PLATED STEEL	36.00	4.00	Brand#41	ECONOMY POLISHED STEEL	9.00	4.00
Brand#35	PROMO PLATED TIN	19.00	4.00	Brand#41	ECONOMY POLISHED STEEL	19.00	4.00
Brand#35	PROMO POLISHED BRASS	9.00	4.00	Brand#41	ECONOMY POLISHED STEEL	45.00	4.00
Brand#35	PROMO POLISHED BRASS	36.00	4.00	Brand#41	ECONOMY POLISHED TIN	19.00	4.00
Brand#35	PROMO POLISHED NICKEL	36.00	4.00	Brand#41	LARGE ANODIZED BRASS	14.00	4.00
Brand#35	PROMO POLISHED STEEL	23.00	4.00	Brand#41	LARGE ANODIZED BRASS	23.00	4.00
Brand#35	PROMO POLISHED TIN	36.00	4.00	Brand#41	LARGE ANODIZED COPPER	49.00	4.00
Brand#35	PROMO POLISHED TIN	45.00	4.00	Brand#41	LARGE ANODIZED STEEL	3.00	4.00

Brand#41	LARGE ANODIZED STEEL	23.00	4.00	Brand#41	STANDARD ANODIZED TIN	36.00	4.00
Brand#41	LARGE BRUSHED COPPER	23.00	4.00	Brand#41	STANDARD ANODIZED TIN	49.00	4.00
Brand#41	LARGE BRUSHED COPPER	49.00	4.00	Brand#41	STANDARD BRUSHED BRASS	19.00	4.00
Brand#41	LARGE BRUSHED STEEL	19.00	4.00	Brand#41	STANDARD BRUSHED NICKEL	3.00	4.00
Brand#41	LARGE BURNISHED BRASS	45.00	4.00	Brand#41	STANDARD BRUSHED NICKEL	9.00	4.00
Brand#41	LARGE BURNISHED COPPER	3.00	4.00	Brand#41	STANDARD BRUSHED STEEL	45.00	4.00
Brand#41	LARGE BURNISHED NICKEL	23.00	4.00	Brand#41	STANDARD BRUSHED TIN	45.00	4.00
Brand#41	LARGE BURNISHED TIN	9.00	4.00	Brand#41	STANDARD BURNISHED BRASS	23.00	4.00
Brand#41	LARGE PLATED NICKEL	3.00	4.00	Brand#41	STANDARD BURNISHED BRASS	36.00	4.00
Brand#41	LARGE PLATED NICKEL	23.00	4.00	Brand#41	STANDARD BURNISHED COPPER	49.00	4.00
Brand#41	LARGE PLATED STEEL	9.00	4.00	Brand#41	STANDARD BURNISHED STEEL	45.00	4.00
Brand#41	LARGE PLATED STEEL	36.00	4.00	Brand#41	STANDARD PLATED BRASS	45.00	4.00
Brand#41	LARGE PLATED TIN	9.00	4.00	Brand#41	STANDARD PLATED NICKEL	14.00	4.00
Brand#41	LARGE POLISHED BRASS	36.00	4.00	Brand#41	STANDARD PLATED STEEL	45.00	4.00
Brand#41	LARGE POLISHED COPPER	19.00	4.00	Brand#41	STANDARD PLATED TIN	49.00	4.00
Brand#41	LARGE POLISHED COPPER	49.00	4.00	Brand#41	STANDARD POLISHED STEEL	9.00	4.00
Brand#41	LARGE POLISHED NICKEL	36.00	4.00	Brand#41	STANDARD POLISHED STEEL	19.00	4.00
Brand#41	LARGE POLISHED STEEL	14.00	4.00	Brand#41	STANDARD POLISHED TIN	45.00	4.00
Brand#41	MEDIUM ANODIZED BRASS	9.00	4.00	Brand#42	ECONOMY ANODIZED NICKEL	19.00	4.00
Brand#41	MEDIUM ANODIZED COPPER	14.00	4.00	Brand#42	ECONOMY BRUSHED BRASS	14.00	4.00
Brand#41	MEDIUM ANODIZED NICKEL	3.00	4.00	Brand#42	ECONOMY BRUSHED COPPER	3.00	4.00
Brand#41	MEDIUM ANODIZED NICKEL	9.00	4.00	Brand#42	ECONOMY BRUSHED COPPER	14.00	4.00
Brand#41	MEDIUM ANODIZED STEEL	14.00	4.00	Brand#42	ECONOMY BRUSHED NICKEL	14.00	4.00
Brand#41	MEDIUM BRUSHED COPPER	3.00	4.00	Brand#42	ECONOMY BRUSHED STEEL	14.00	4.00
Brand#41	MEDIUM BRUSHED TIN	9.00	4.00	Brand#42	ECONOMY BRUSHED TIN	19.00	4.00
Brand#41	MEDIUM BURNISHED COPPER	23.00	4.00	Brand#42	ECONOMY BRUSHED TIN	49.00	4.00
Brand#41	MEDIUM BURNISHED STEEL	9.00	4.00	Brand#42	ECONOMY BURNISHED BRASS	19.00	4.00
Brand#41	MEDIUM BURNISHED STEEL	45.00	4.00	Brand#42	ECONOMY BURNISHED COPPER	23.00	4.00
Brand#41	MEDIUM BURNISHED TIN	3.00	4.00	Brand#42	ECONOMY BURNISHED NICKEL	14.00	4.00
Brand#41	MEDIUM PLATED BRASS	19.00	4.00	Brand#42	ECONOMY BURNISHED TIN	14.00	4.00
Brand#41	MEDIUM PLATED BRASS	45.00	4.00	Brand#42	ECONOMY PLATED COPPER	23.00	4.00
Brand#41	MEDIUM PLATED COPPER	19.00	4.00	Brand#42	ECONOMY POLISHED BRASS	3.00	4.00
Brand#41	MEDIUM PLATED STEEL	19.00	4.00	Brand#42	ECONOMY POLISHED COPPER	9.00	4.00
Brand#41	MEDIUM PLATED STEEL	23.00	4.00	Brand#42	ECONOMY POLISHED STEEL	9.00	4.00
Brand#41	PROMO ANODIZED BRASS	19.00	4.00	Brand#42	ECONOMY POLISHED STEEL	36.00	4.00
Brand#41	PROMO ANODIZED COPPER	9.00	4.00	Brand#42	ECONOMY POLISHED TIN	14.00	4.00
Brand#41	PROMO ANODIZED NICKEL	9.00	4.00	Brand#42	LARGE ANODIZED BRASS	49.00	4.00
Brand#41	PROMO BRUSHED BRASS	14.00	4.00	Brand#42	LARGE ANODIZED COPPER	14.00	4.00
Brand#41	PROMO BRUSHED COPPER	36.00	4.00	Brand#42	LARGE ANODIZED COPPER	49.00	4.00
Brand#41	PROMO BRUSHED NICKEL	14.00	4.00	Brand#42	LARGE ANODIZED NICKEL	45.00	4.00
Brand#41	PROMO BURNISHED BRASS	49.00	4.00	Brand#42	LARGE ANODIZED NICKEL	49.00	4.00
Brand#41	PROMO BURNISHED TIN	3.00	4.00	Brand#42	LARGE ANODIZED TIN	45.00	4.00
Brand#41	PROMO PLATED NICKEL	14.00	4.00	Brand#42	LARGE BRUSHED BRASS	49.00	4.00
Brand#41	PROMO PLATED NICKEL	45.00	4.00	Brand#42	LARGE BURNISHED BRASS	45.00	4.00
Brand#41	PROMO PLATED STEEL	3.00	4.00	Brand#42	LARGE BURNISHED BRASS	49.00	4.00
Brand#41	PROMO PLATED TIN	3.00	4.00	Brand#42	LARGE BURNISHED COPPER	9.00	4.00
Brand#41	PROMO POLISHED COPPER	23.00	4.00	Brand#42	LARGE BURNISHED TIN	9.00	4.00
Brand#41	SMALL ANODIZED BRASS	3.00	4.00	Brand#42	LARGE PLATED BRASS	45.00	4.00
Brand#41	SMALL ANODIZED BRASS	14.00	4.00	Brand#42	LARGE PLATED COPPER	9.00	4.00
Brand#41	SMALL ANODIZED STEEL	45.00	4.00	Brand#42	LARGE PLATED NICKEL	36.00	4.00
Brand#41	SMALL ANODIZED TIN	9.00	4.00	Brand#42	LARGE PLATED TIN	23.00	4.00
Brand#41	SMALL BRUSHED TIN	19.00	4.00	Brand#42	LARGE POLISHED BRASS	9.00	4.00
Brand#41	SMALL BURNISHED COPPER	9.00	4.00	Brand#42	LARGE POLISHED NICKEL	3.00	4.00
Brand#41	SMALL BURNISHED NICKEL	3.00	4.00	Brand#42	LARGE POLISHED NICKEL	23.00	4.00
Brand#41	SMALL BURNISHED TIN	45.00	4.00	Brand#42	LARGE POLISHED STEEL	9.00	4.00
Brand#41	SMALL PLATED COPPER	14.00	4.00	Brand#42	MEDIUM ANODIZED BRASS	23.00	4.00
Brand#41	SMALL PLATED COPPER	36.00	4.00	Brand#42	MEDIUM ANODIZED COPPER	19.00	4.00
Brand#41	SMALL PLATED COPPER	49.00	4.00	Brand#42	MEDIUM ANODIZED NICKEL	14.00	4.00
Brand#41	SMALL PLATED TIN	19.00	4.00	Brand#42	MEDIUM ANODIZED NICKEL	19.00	4.00
Brand#41	SMALL POLISHED COPPER	14.00	4.00	Brand#42	MEDIUM ANODIZED NICKEL	23.00	4.00
Brand#41	SMALL POLISHED COPPER	19.00	4.00	Brand#42	MEDIUM ANODIZED STEEL	9.00	4.00
Brand#41	SMALL POLISHED COPPER	36.00	4.00	Brand#42	MEDIUM ANODIZED STEEL	14.00	4.00
Brand#41	SMALL POLISHED TIN	45.00	4.00	Brand#42	MEDIUM ANODIZED STEEL	23.00	4.00
Brand#41	STANDARD ANODIZED COPPER	19.00	4.00	Brand#42	MEDIUM ANODIZED TIN	14.00	4.00
Brand#41	STANDARD ANODIZED NICKEL	9.00	4.00	Brand#42	MEDIUM ANODIZED TIN	19.00	4.00
Brand#41	STANDARD ANODIZED STEEL	49.00	4.00	Brand#42	MEDIUM BRUSHED COPPER	45.00	4.00
Brand#41	STANDARD ANODIZED TIN	9.00	4.00	Brand#42	MEDIUM BRUSHED COPPER	49.00	4.00

Brand#42	MEDIUM BRUSHED STEEL	36.00	4.00	Brand#43	ECONOMY BRUSHED COPPER	45.00	4.00
Brand#42	MEDIUM BURNISHED COPPER	49.00	4.00	Brand#43	ECONOMY BRUSHED NICKEL	9.00	4.00
Brand#42	MEDIUM BURNISHED TIN	3.00	4.00	Brand#43	ECONOMY BURNISHED NICKEL	9.00	4.00
Brand#42	MEDIUM BURNISHED TIN	49.00	4.00	Brand#43	ECONOMY BURNISHED TIN	19.00	4.00
Brand#42	MEDIUM PLATED NICKEL	45.00	4.00	Brand#43	ECONOMY PLATED COPPER	36.00	4.00
Brand#42	MEDIUM PLATED STEEL	3.00	4.00	Brand#43	ECONOMY PLATED STEEL	9.00	4.00
Brand#42	MEDIUM PLATED STEEL	23.00	4.00	Brand#43	ECONOMY PLATED TIN	14.00	4.00
Brand#42	MEDIUM PLATED STEEL	45.00	4.00	Brand#43	ECONOMY PLATED TIN	19.00	4.00
Brand#42	PROMO ANODIZED NICKEL	3.00	4.00	Brand#43	ECONOMY PLATED TIN	49.00	4.00
Brand#42	PROMO ANODIZED NICKEL	19.00	4.00	Brand#43	ECONOMY POLISHED COPPER	19.00	4.00
Brand#42	PROMO ANODIZED STEEL	49.00	4.00	Brand#43	ECONOMY POLISHED NICKEL	36.00	4.00
Brand#42	PROMO BRUSHED COPPER	45.00	4.00	Brand#43	ECONOMY POLISHED TIN	14.00	4.00
Brand#42	PROMO BRUSHED STEEL	19.00	4.00	Brand#43	ECONOMY POLISHED TIN	45.00	4.00
Brand#42	PROMO BRUSHED TIN	45.00	4.00	Brand#43	LARGE ANODIZED BRASS	14.00	4.00
Brand#42	PROMO BURNISHED COPPER	45.00	4.00	Brand#43	LARGE ANODIZED BRASS	36.00	4.00
Brand#42	PROMO BURNISHED NICKEL	3.00	4.00	Brand#43	LARGE ANODIZED COPPER	45.00	4.00
Brand#42	PROMO BURNISHED STEEL	9.00	4.00	Brand#43	LARGE BRUSHED COPPER	3.00	4.00
Brand#42	PROMO BURNISHED TIN	49.00	4.00	Brand#43	LARGE BRUSHED NICKEL	14.00	4.00
Brand#42	PROMO PLATED BRASS	45.00	4.00	Brand#43	LARGE BRUSHED NICKEL	19.00	4.00
Brand#42	PROMO PLATED NICKEL	23.00	4.00	Brand#43	LARGE BRUSHED NICKEL	45.00	4.00
Brand#42	PROMO PLATED STEEL	19.00	4.00	Brand#43	LARGE BRUSHED NICKEL	49.00	4.00
Brand#42	PROMO PLATED STEEL	45.00	4.00	Brand#43	LARGE BURNISHED COPPER	3.00	4.00
Brand#42	PROMO POLISHED COPPER	36.00	4.00	Brand#43	LARGE BURNISHED TIN	23.00	4.00
Brand#42	PROMO POLISHED NICKEL	3.00	4.00	Brand#43	LARGE BURNISHED TIN	45.00	4.00
Brand#42	SMALL ANODIZED BRASS	23.00	4.00	Brand#43	LARGE PLATED BRASS	45.00	4.00
Brand#42	SMALL ANODIZED COPPER	14.00	4.00	Brand#43	LARGE PLATED STEEL	14.00	4.00
Brand#42	SMALL ANODIZED COPPER	19.00	4.00	Brand#43	LARGE PLATED TIN	36.00	4.00
Brand#42	SMALL ANODIZED NICKEL	23.00	4.00	Brand#43	LARGE PLATED TIN	45.00	4.00
Brand#42	SMALL BRUSHED TIN	49.00	4.00	Brand#43	LARGE POLISHED BRASS	9.00	4.00
Brand#42	SMALL BURNISHED BRASS	3.00	4.00	Brand#43	LARGE POLISHED COPPER	9.00	4.00
Brand#42	SMALL BURNISHED BRASS	36.00	4.00	Brand#43	LARGE POLISHED COPPER	19.00	4.00
Brand#42	SMALL BURNISHED COPPER	9.00	4.00	Brand#43	LARGE POLISHED STEEL	14.00	4.00
Brand#42	SMALL BURNISHED NICKEL	9.00	4.00	Brand#43	LARGE POLISHED TIN	45.00	4.00
Brand#42	SMALL BURNISHED TIN	9.00	4.00	Brand#43	MEDIUM ANODIZED BRASS	14.00	4.00
Brand#42	SMALL PLATED NICKEL	9.00	4.00	Brand#43	MEDIUM ANODIZED COPPER	36.00	4.00
Brand#42	SMALL PLATED TIN	36.00	4.00	Brand#43	MEDIUM ANODIZED COPPER	49.00	4.00
Brand#42	SMALL POLISHED BRASS	3.00	4.00	Brand#43	MEDIUM ANODIZED STEEL	19.00	4.00
Brand#42	SMALL POLISHED COPPER	36.00	4.00	Brand#43	MEDIUM ANODIZED STEEL	36.00	4.00
Brand#42	SMALL POLISHED NICKEL	23.00	4.00	Brand#43	MEDIUM BRUSHED BRASS	9.00	4.00
Brand#42	SMALL POLISHED STEEL	49.00	4.00	Brand#43	MEDIUM BRUSHED BRASS	49.00	4.00
Brand#42	SMALL POLISHED TIN	3.00	4.00	Brand#43	MEDIUM BRUSHED COPPER	3.00	4.00
Brand#42	STANDARD ANODIZED BRASS	49.00	4.00	Brand#43	MEDIUM BRUSHED NICKEL	9.00	4.00
Brand#42	STANDARD ANODIZED COPPER	49.00	4.00	Brand#43	MEDIUM BRUSHED STEEL	23.00	4.00
Brand#42	STANDARD ANODIZED NICKEL	36.00	4.00	Brand#43	MEDIUM BURNISHED COPPER	14.00	4.00
Brand#42	STANDARD ANODIZED NICKEL	45.00	4.00	Brand#43	MEDIUM BURNISHED COPPER	45.00	4.00
Brand#42	STANDARD BRUSHED NICKEL	23.00	4.00	Brand#43	MEDIUM BURNISHED TIN	23.00	4.00
Brand#42	STANDARD BURNISHED NICKEL	49.00	4.00	Brand#43	MEDIUM PLATED BRASS	3.00	4.00
Brand#42	STANDARD BURNISHED STEEL	3.00	4.00	Brand#43	MEDIUM PLATED COPPER	14.00	4.00
Brand#42	STANDARD BURNISHED TIN	19.00	4.00	Brand#43	MEDIUM PLATED NICKEL	36.00	4.00
Brand#42	STANDARD PLATED BRASS	19.00	4.00	Brand#43	MEDIUM PLATED NICKEL	45.00	4.00
Brand#42	STANDARD PLATED COPPER	9.00	4.00	Brand#43	MEDIUM PLATED TIN	49.00	4.00
Brand#42	STANDARD PLATED NICKEL	45.00	4.00	Brand#43	PROMO ANODIZED NICKEL	45.00	4.00
Brand#42	STANDARD PLATED STEEL	3.00	4.00	Brand#43	PROMO ANODIZED TIN	14.00	4.00
Brand#42	STANDARD POLISHED BRASS	36.00	4.00	Brand#43	PROMO BRUSHED NICKEL	14.00	4.00
Brand#42	STANDARD POLISHED BRASS	45.00	4.00	Brand#43	PROMO BRUSHED STEEL	14.00	4.00
Brand#42	STANDARD POLISHED COPPER	14.00	4.00	Brand#43	PROMO BRUSHED TIN	45.00	4.00
Brand#42	STANDARD POLISHED NICKEL	45.00	4.00	Brand#43	PROMO BURNISHED BRASS	49.00	4.00
Brand#42	STANDARD POLISHED TIN	9.00	4.00	Brand#43	PROMO BURNISHED NICKEL	9.00	4.00
Brand#42	STANDARD POLISHED TIN	19.00	4.00	Brand#43	PROMO BURNISHED STEEL	3.00	4.00
Brand#42	STANDARD POLISHED TIN	23.00	4.00	Brand#43	PROMO BURNISHED STEEL	36.00	4.00
Brand#42	STANDARD POLISHED TIN	36.00	4.00	Brand#43	PROMO BURNISHED TIN	36.00	4.00
Brand#43	ECONOMY ANODIZED COPPER	19.00	4.00	Brand#43	PROMO PLATED BRASS	19.00	4.00
Brand#43	ECONOMY ANODIZED COPPER	45.00	4.00	Brand#43	PROMO PLATED COPPER	45.00	4.00
Brand#43	ECONOMY ANODIZED NICKEL	3.00	4.00	Brand#43	PROMO PLATED COPPER	49.00	4.00
Brand#43	ECONOMY ANODIZED NICKEL	49.00	4.00	Brand#43	PROMO PLATED TIN	3.00	4.00
Brand#43	ECONOMY ANODIZED STEEL	23.00	4.00	Brand#43	PROMO POLISHED BRASS	19.00	4.00
Brand#43	ECONOMY ANODIZED TIN	49.00	4.00	Brand#43	PROMO POLISHED BRASS	23.00	4.00
Brand#43	ECONOMY BRUSHED BRASS	49.00	4.00	Brand#43	PROMO POLISHED NICKEL	49.00	4.00

Brand#43	PROMO POLISHED STEEL	14.00	4.00	Brand#44	LARGE PLATED NICKEL	14.00	4.00
Brand#43	PROMO POLISHED STEEL	19.00	4.00	Brand#44	LARGE PLATED STEEL	14.00	4.00
Brand#43	PROMO POLISHED STEEL	23.00	4.00	Brand#44	LARGE PLATED TIN	19.00	4.00
Brand#43	PROMO POLISHED STEEL	36.00	4.00	Brand#44	LARGE PLATED TIN	23.00	4.00
Brand#43	SMALL ANODIZED BRASS	19.00	4.00	Brand#44	LARGE POLISHED STEEL	23.00	4.00
Brand#43	SMALL ANODIZED NICKEL	9.00	4.00	Brand#44	LARGE POLISHED STEEL	49.00	4.00
Brand#43	SMALL BRUSHED NICKEL	3.00	4.00	Brand#44	MEDIUM ANODIZED COPPER	45.00	4.00
Brand#43	SMALL BRUSHED NICKEL	9.00	4.00	Brand#44	MEDIUM ANODIZED NICKEL	45.00	4.00
Brand#43	SMALL BURNISHED BRASS	49.00	4.00	Brand#44	MEDIUM BRUSHED BRASS	49.00	4.00
Brand#43	SMALL BURNISHED STEEL	23.00	4.00	Brand#44	MEDIUM BRUSHED COPPER	3.00	4.00
Brand#43	SMALL PLATED BRASS	14.00	4.00	Brand#44	MEDIUM BRUSHED COPPER	45.00	4.00
Brand#43	SMALL PLATED BRASS	36.00	4.00	Brand#44	MEDIUM BRUSHED STEEL	19.00	4.00
Brand#43	SMALL PLATED COPPER	23.00	4.00	Brand#44	MEDIUM BRUSHED TIN	49.00	4.00
Brand#43	SMALL PLATED COPPER	49.00	4.00	Brand#44	MEDIUM BURNISHED COPPER	45.00	4.00
Brand#43	SMALL PLATED NICKEL	36.00	4.00	Brand#44	MEDIUM BURNISHED NICKEL	23.00	4.00
Brand#43	SMALL PLATED NICKEL	49.00	4.00	Brand#44	MEDIUM BURNISHED TIN	23.00	4.00
Brand#43	SMALL PLATED STEEL	14.00	4.00	Brand#44	MEDIUM PLATED COPPER	14.00	4.00
Brand#43	SMALL PLATED TIN	49.00	4.00	Brand#44	PROMO ANODIZED COPPER	23.00	4.00
Brand#43	SMALL POLISHED STEEL	19.00	4.00	Brand#44	PROMO ANODIZED STEEL	36.00	4.00
Brand#43	STANDARD ANODIZED BRASS	3.00	4.00	Brand#44	PROMO BRUSHED COPPER	23.00	4.00
Brand#43	STANDARD ANODIZED COPPER	49.00	4.00	Brand#44	PROMO BRUSHED COPPER	36.00	4.00
Brand#43	STANDARD ANODIZED NICKEL	14.00	4.00	Brand#44	PROMO BRUSHED TIN	19.00	4.00
Brand#43	STANDARD BRUSHED TIN	14.00	4.00	Brand#44	PROMO PLATED BRASS	3.00	4.00
Brand#43	STANDARD BURNISHED BRASS	23.00	4.00	Brand#44	PROMO PLATED COPPER	36.00	4.00
Brand#43	STANDARD BURNISHED STEEL	19.00	4.00	Brand#44	PROMO PLATED STEEL	3.00	4.00
Brand#43	STANDARD BURNISHED STEEL	23.00	4.00	Brand#44	PROMO PLATED STEEL	36.00	4.00
Brand#43	STANDARD PLATED BRASS	9.00	4.00	Brand#44	PROMO PLATED STEEL	49.00	4.00
Brand#43	STANDARD PLATED BRASS	19.00	4.00	Brand#44	PROMO POLISHED BRASS	3.00	4.00
Brand#43	STANDARD PLATED BRASS	49.00	4.00	Brand#44	PROMO POLISHED BRASS	19.00	4.00
Brand#43	STANDARD PLATED COPPER	36.00	4.00	Brand#44	PROMO POLISHED COPPER	45.00	4.00
Brand#43	STANDARD PLATED NICKEL	14.00	4.00	Brand#44	PROMO POLISHED STEEL	36.00	4.00
Brand#43	STANDARD PLATED NICKEL	19.00	4.00	Brand#44	PROMO POLISHED TIN	9.00	4.00
Brand#43	STANDARD PLATED TIN	14.00	4.00	Brand#44	SMALL ANODIZED COPPER	23.00	4.00
Brand#43	STANDARD POLISHED BRASS	23.00	4.00	Brand#44	SMALL ANODIZED STEEL	23.00	4.00
Brand#43	STANDARD POLISHED TIN	9.00	4.00	Brand#44	SMALL ANODIZED TIN	45.00	4.00
Brand#44	ECONOMY ANODIZED BRASS	3.00	4.00	Brand#44	SMALL BRUSHED COPPER	14.00	4.00
Brand#44	ECONOMY ANODIZED BRASS	45.00	4.00	Brand#44	SMALL BRUSHED STEEL	45.00	4.00
Brand#44	ECONOMY ANODIZED NICKEL	36.00	4.00	Brand#44	SMALL BURNISHED COPPER	14.00	4.00
Brand#44	ECONOMY ANODIZED STEEL	19.00	4.00	Brand#44	SMALL BURNISHED COPPER	49.00	4.00
Brand#44	ECONOMY BRUSHED COPPER	23.00	4.00	Brand#44	SMALL BURNISHED NICKEL	14.00	4.00
Brand#44	ECONOMY BRUSHED TIN	49.00	4.00	Brand#44	SMALL BURNISHED STEEL	23.00	4.00
Brand#44	ECONOMY BURNISHED COPPER	19.00	4.00	Brand#44	SMALL BURNISHED TIN	49.00	4.00
Brand#44	ECONOMY BURNISHED STEEL	45.00	4.00	Brand#44	SMALL PLATED BRASS	36.00	4.00
Brand#44	ECONOMY PLATED STEEL	19.00	4.00	Brand#44	SMALL PLATED COPPER	19.00	4.00
Brand#44	ECONOMY PLATED STEEL	23.00	4.00	Brand#44	SMALL PLATED NICKEL	3.00	4.00
Brand#44	ECONOMY PLATED TIN	23.00	4.00	Brand#44	SMALL POLISHED COPPER	3.00	4.00
Brand#44	ECONOMY POLISHED BRASS	23.00	4.00	Brand#44	SMALL POLISHED COPPER	49.00	4.00
Brand#44	ECONOMY POLISHED COPPER	9.00	4.00	Brand#44	SMALL POLISHED STEEL	3.00	4.00
Brand#44	ECONOMY POLISHED COPPER	45.00	4.00	Brand#44	STANDARD ANODIZED BRASS	3.00	4.00
Brand#44	ECONOMY POLISHED NICKEL	14.00	4.00	Brand#44	STANDARD ANODIZED COPPER	3.00	4.00
Brand#44	ECONOMY POLISHED NICKEL	23.00	4.00	Brand#44	STANDARD ANODIZED NICKEL	3.00	4.00
Brand#44	ECONOMY POLISHED STEEL	49.00	4.00	Brand#44	STANDARD ANODIZED NICKEL	36.00	4.00
Brand#44	ECONOMY POLISHED TIN	23.00	4.00	Brand#44	STANDARD ANODIZED STEEL	14.00	4.00
Brand#44	ECONOMY POLISHED TIN	36.00	4.00	Brand#44	STANDARD ANODIZED TIN	3.00	4.00
Brand#44	LARGE ANODIZED BRASS	19.00	4.00	Brand#44	STANDARD ANODIZED TIN	9.00	4.00
Brand#44	LARGE ANODIZED TIN	3.00	4.00	Brand#44	STANDARD ANODIZED TIN	36.00	4.00
Brand#44	LARGE ANODIZED TIN	14.00	4.00	Brand#44	STANDARD BRUSHED COPPER	36.00	4.00
Brand#44	LARGE BRUSHED TIN	3.00	4.00	Brand#44	STANDARD BRUSHED COPPER	45.00	4.00
Brand#44	LARGE BRUSHED TIN	23.00	4.00	Brand#44	STANDARD BRUSHED TIN	9.00	4.00
Brand#44	LARGE BURNISHED BRASS	23.00	4.00	Brand#44	STANDARD BRUSHED TIN	49.00	4.00
Brand#44	LARGE BURNISHED BRASS	49.00	4.00	Brand#44	STANDARD BURNISHED COPPER	9.00	4.00
Brand#44	LARGE BURNISHED COPPER	3.00	4.00	Brand#44	STANDARD BURNISHED STEEL	23.00	4.00
Brand#44	LARGE BURNISHED COPPER	19.00	4.00	Brand#44	STANDARD PLATED BRASS	14.00	4.00
Brand#44	LARGE BURNISHED COPPER	36.00	4.00	Brand#44	STANDARD PLATED BRASS	23.00	4.00
Brand#44	LARGE BURNISHED TIN	14.00	4.00	Brand#44	STANDARD PLATED BRASS	49.00	4.00
Brand#44	LARGE PLATED BRASS	9.00	4.00	Brand#44	STANDARD PLATED COPPER	14.00	4.00
Brand#44	LARGE PLATED BRASS	49.00	4.00	Brand#44	STANDARD POLISHED NICKEL	19.00	4.00

Brand#44	STANDARD POLISHED TIN	9.00	4.00	Brand#51	SMALL PLATED COPPER	49.00	4.00
Brand#51	ECONOMY ANODIZED BRASS	9.00	4.00	Brand#51	SMALL PLATED NICKEL	3.00	4.00
Brand#51	ECONOMY ANODIZED BRASS	23.00	4.00	Brand#51	SMALL PLATED NICKEL	14.00	4.00
Brand#51	ECONOMY ANODIZED NICKEL	3.00	4.00	Brand#51	SMALL PLATED STEEL	45.00	4.00
Brand#51	ECONOMY ANODIZED NICKEL	23.00	4.00	Brand#51	SMALL POLISHED NICKEL	14.00	4.00
Brand#51	ECONOMY ANODIZED STEEL	19.00	4.00	Brand#51	SMALL POLISHED NICKEL	23.00	4.00
Brand#51	ECONOMY ANODIZED STEEL	23.00	4.00	Brand#51	SMALL POLISHED STEEL	3.00	4.00
Brand#51	ECONOMY ANODIZED STEEL	49.00	4.00	Brand#51	SMALL POLISHED STEEL	19.00	4.00
Brand#51	ECONOMY BRUSHED BRASS	3.00	4.00	Brand#51	SMALL POLISHED STEEL	49.00	4.00
Brand#51	ECONOMY BRUSHED BRASS	49.00	4.00	Brand#51	STANDARD ANODIZED NICKEL	3.00	4.00
Brand#51	ECONOMY BRUSHED NICKEL	14.00	4.00	Brand#51	STANDARD ANODIZED NICKEL	49.00	4.00
Brand#51	ECONOMY BRUSHED STEEL	45.00	4.00	Brand#51	STANDARD BRUSHED BRASS	3.00	4.00
Brand#51	ECONOMY BRUSHED TIN	36.00	4.00	Brand#51	STANDARD BRUSHED COPPER	3.00	4.00
Brand#51	ECONOMY BURNISHED BRASS	14.00	4.00	Brand#51	STANDARD BRUSHED NICKEL	19.00	4.00
Brand#51	ECONOMY BURNISHED COPPER	45.00	4.00	Brand#51	STANDARD BRUSHED STEEL	36.00	4.00
Brand#51	ECONOMY PLATED NICKEL	49.00	4.00	Brand#51	STANDARD BURNISHED COPPER	19.00	4.00
Brand#51	ECONOMY PLATED TIN	36.00	4.00	Brand#51	STANDARD BURNISHED NICKEL	49.00	4.00
Brand#51	ECONOMY POLISHED COPPER	9.00	4.00	Brand#51	STANDARD BURNISHED STEEL	23.00	4.00
Brand#51	ECONOMY POLISHED STEEL	14.00	4.00	Brand#51	STANDARD BURNISHED STEEL	36.00	4.00
Brand#51	ECONOMY POLISHED STEEL	49.00	4.00	Brand#51	STANDARD BURNISHED TIN	45.00	4.00
Brand#51	LARGE ANODIZED COPPER	9.00	4.00	Brand#51	STANDARD PLATED BRASS	36.00	4.00
Brand#51	LARGE ANODIZED COPPER	49.00	4.00	Brand#51	STANDARD PLATED BRASS	49.00	4.00
Brand#51	LARGE ANODIZED NICKEL	14.00	4.00	Brand#51	STANDARD PLATED COPPER	14.00	4.00
Brand#51	LARGE ANODIZED STEEL	36.00	4.00	Brand#51	STANDARD PLATED COPPER	23.00	4.00
Brand#51	LARGE BRUSHED NICKEL	3.00	4.00	Brand#51	STANDARD POLISHED BRASS	14.00	4.00
Brand#51	LARGE BRUSHED NICKEL	9.00	4.00	Brand#51	STANDARD POLISHED BRASS	45.00	4.00
Brand#51	LARGE BURNISHED BRASS	19.00	4.00	Brand#51	STANDARD POLISHED STEEL	36.00	4.00
Brand#51	LARGE BURNISHED BRASS	36.00	4.00	Brand#51	STANDARD POLISHED STEEL	49.00	4.00
Brand#51	LARGE BURNISHED COPPER	14.00	4.00	Brand#51	STANDARD POLISHED TIN	45.00	4.00
Brand#51	LARGE BURNISHED NICKEL	14.00	4.00	Brand#52	ECONOMY ANODIZED BRASS	14.00	4.00
Brand#51	LARGE PLATED BRASS	36.00	4.00	Brand#52	ECONOMY ANODIZED BRASS	23.00	4.00
Brand#51	LARGE POLISHED COPPER	14.00	4.00	Brand#52	ECONOMY ANODIZED COPPER	36.00	4.00
Brand#51	LARGE POLISHED NICKEL	23.00	4.00	Brand#52	ECONOMY ANODIZED NICKEL	49.00	4.00
Brand#51	LARGE POLISHED NICKEL	36.00	4.00	Brand#52	ECONOMY ANODIZED STEEL	19.00	4.00
Brand#51	LARGE POLISHED STEEL	19.00	4.00	Brand#52	ECONOMY BRUSHED COPPER	49.00	4.00
Brand#51	MEDIUM ANODIZED COPPER	9.00	4.00	Brand#52	ECONOMY BURNISHED BRASS	36.00	4.00
Brand#51	MEDIUM ANODIZED STEEL	3.00	4.00	Brand#52	ECONOMY BURNISHED COPPER	19.00	4.00
Brand#51	MEDIUM BRUSHED BRASS	36.00	4.00	Brand#52	ECONOMY BURNISHED COPPER	45.00	4.00
Brand#51	MEDIUM BRUSHED BRASS	45.00	4.00	Brand#52	ECONOMY BURNISHED NICKEL	19.00	4.00
Brand#51	MEDIUM BRUSHED STEEL	3.00	4.00	Brand#52	ECONOMY BURNISHED STEEL	36.00	4.00
Brand#51	MEDIUM BURNISHED NICKEL	3.00	4.00	Brand#52	ECONOMY PLATED TIN	14.00	4.00
Brand#51	MEDIUM BURNISHED NICKEL	36.00	4.00	Brand#52	ECONOMY PLATED TIN	23.00	4.00
Brand#51	MEDIUM BURNISHED STEEL	14.00	4.00	Brand#52	ECONOMY POLISHED BRASS	23.00	4.00
Brand#51	MEDIUM BURNISHED TIN	9.00	4.00	Brand#52	ECONOMY POLISHED BRASS	45.00	4.00
Brand#51	MEDIUM PLATED STEEL	19.00	4.00	Brand#52	ECONOMY POLISHED NICKEL	36.00	4.00
Brand#51	MEDIUM PLATED TIN	3.00	4.00	Brand#52	ECONOMY POLISHED STEEL	49.00	4.00
Brand#51	PROMO ANODIZED NICKEL	14.00	4.00	Brand#52	LARGE ANODIZED COPPER	14.00	4.00
Brand#51	PROMO ANODIZED STEEL	23.00	4.00	Brand#52	LARGE ANODIZED NICKEL	3.00	4.00
Brand#51	PROMO ANODIZED TIN	19.00	4.00	Brand#52	LARGE ANODIZED NICKEL	45.00	4.00
Brand#51	PROMO BRUSHED BRASS	23.00	4.00	Brand#52	LARGE ANODIZED TIN	45.00	4.00
Brand#51	PROMO BRUSHED COPPER	45.00	4.00	Brand#52	LARGE BRUSHED COPPER	19.00	4.00
Brand#51	PROMO BRUSHED STEEL	45.00	4.00	Brand#52	LARGE BRUSHED COPPER	3.00	4.00
Brand#51	PROMO BRUSHED TIN	9.00	4.00	Brand#52	LARGE BRUSHED NICKEL	19.00	4.00
Brand#51	PROMO BURNISHED BRASS	19.00	4.00	Brand#52	LARGE BRUSHED NICKEL	23.00	4.00
Brand#51	PROMO BURNISHED BRASS	23.00	4.00	Brand#52	LARGE BRUSHED STEEL	49.00	4.00
Brand#51	PROMO BURNISHED NICKEL	14.00	4.00	Brand#52	LARGE BRUSHED TIN	14.00	4.00
Brand#51	PROMO PLATED BRASS	3.00	4.00	Brand#52	LARGE BURNISHED NICKEL	9.00	4.00
Brand#51	PROMO PLATED BRASS	23.00	4.00	Brand#52	LARGE BURNISHED TIN	23.00	4.00
Brand#51	PROMO PLATED TIN	19.00	4.00	Brand#52	LARGE BURNISHED TIN	45.00	4.00
Brand#51	PROMO PLATED TIN	23.00	4.00	Brand#52	LARGE PLATED BRASS	14.00	4.00
Brand#51	PROMO POLISHED BRASS	23.00	4.00	Brand#52	LARGE PLATED COPPER	14.00	4.00
Brand#51	PROMO POLISHED COPPER	9.00	4.00	Brand#52	LARGE PLATED COPPER	19.00	4.00
Brand#51	PROMO POLISHED NICKEL	9.00	4.00	Brand#52	LARGE PLATED NICKEL	45.00	4.00
Brand#51	PROMO POLISHED STEEL	49.00	4.00	Brand#52	LARGE PLATED STEEL	9.00	4.00
Brand#51	SMALL ANODIZED STEEL	14.00	4.00	Brand#52	LARGE PLATED TIN	9.00	4.00
Brand#51	SMALL BRUSHED BRASS	23.00	4.00	Brand#52	LARGE POLISHED NICKEL	19.00	4.00
Brand#51	SMALL BRUSHED TIN	19.00	4.00	Brand#52	LARGE POLISHED NICKEL	23.00	4.00
Brand#51	SMALL BURNISHED NICKEL	23.00	4.00	Brand#52	LARGE POLISHED NICKEL	36.00	4.00

Brand#52	LARGE POLISHED TIN	9.00	4.00	Brand#52	STANDARD BURNISHED TIN	19.00	4.00
Brand#52	MEDIUM ANODIZED COPPER	36.00	4.00	Brand#52	STANDARD PLATED NICKEL	36.00	4.00
Brand#52	MEDIUM ANODIZED STEEL	14.00	4.00	Brand#52	STANDARD PLATED STEEL	36.00	4.00
Brand#52	MEDIUM ANODIZED TIN	3.00	4.00	Brand#52	STANDARD POLISHED BRASS	36.00	4.00
Brand#52	MEDIUM ANODIZED TIN	49.00	4.00	Brand#52	STANDARD POLISHED COPPER	45.00	4.00
Brand#52	MEDIUM BRUSHED COPPER	9.00	4.00	Brand#52	STANDARD POLISHED STEEL	19.00	4.00
Brand#52	MEDIUM BRUSHED NICKEL	9.00	4.00	Brand#52	STANDARD POLISHED TIN	19.00	4.00
Brand#52	MEDIUM BRUSHED STEEL	23.00	4.00	Brand#53	ECONOMY ANODIZED BRASS	45.00	4.00
Brand#52	MEDIUM BRUSHED STEEL	49.00	4.00	Brand#53	ECONOMY ANODIZED COPPER	9.00	4.00
Brand#52	MEDIUM BURNISHED STEEL	23.00	4.00	Brand#53	ECONOMY ANODIZED NICKEL	3.00	4.00
Brand#52	MEDIUM BURNISHED TIN	45.00	4.00	Brand#53	ECONOMY ANODIZED NICKEL	19.00	4.00
Brand#52	MEDIUM BURNISHED TIN	49.00	4.00	Brand#53	ECONOMY ANODIZED STEEL	45.00	4.00
Brand#52	MEDIUM PLATED BRASS	36.00	4.00	Brand#53	ECONOMY ANODIZED TIN	14.00	4.00
Brand#52	MEDIUM PLATED STEEL	9.00	4.00	Brand#53	ECONOMY ANODIZED TIN	36.00	4.00
Brand#52	MEDIUM PLATED STEEL	49.00	4.00	Brand#53	ECONOMY BRUSHED TIN	45.00	4.00
Brand#52	MEDIUM PLATED TIN	9.00	4.00	Brand#53	ECONOMY BURNISHED BRASS	14.00	4.00
Brand#52	MEDIUM PLATED TIN	49.00	4.00	Brand#53	ECONOMY BURNISHED COPPER	45.00	4.00
Brand#52	PROMO ANODIZED BRASS	9.00	4.00	Brand#53	ECONOMY BURNISHED NICKEL	3.00	4.00
Brand#52	PROMO ANODIZED BRASS	23.00	4.00	Brand#53	ECONOMY BURNISHED NICKEL	49.00	4.00
Brand#52	PROMO ANODIZED BRASS	36.00	4.00	Brand#53	ECONOMY BURNISHED TIN	45.00	4.00
Brand#52	PROMO ANODIZED NICKEL	45.00	4.00	Brand#53	ECONOMY PLATED BRASS	3.00	4.00
Brand#52	PROMO ANODIZED STEEL	36.00	4.00	Brand#53	ECONOMY PLATED NICKEL	14.00	4.00
Brand#52	PROMO BRUSHED COPPER	3.00	4.00	Brand#53	ECONOMY PLATED STEEL	23.00	4.00
Brand#52	PROMO BRUSHED NICKEL	3.00	4.00	Brand#53	ECONOMY PLATED STEEL	36.00	4.00
Brand#52	PROMO BRUSHED NICKEL	49.00	4.00	Brand#53	ECONOMY POLISHED TIN	36.00	4.00
Brand#52	PROMO BRUSHED STEEL	14.00	4.00	Brand#53	LARGE ANODIZED NICKEL	49.00	4.00
Brand#52	PROMO BRUSHED TIN	3.00	4.00	Brand#53	LARGE ANODIZED STEEL	19.00	4.00
Brand#52	PROMO BRUSHED TIN	19.00	4.00	Brand#53	LARGE BRUSHED COPPER	3.00	4.00
Brand#52	PROMO BRUSHED TIN	36.00	4.00	Brand#53	LARGE BRUSHED COPPER	14.00	4.00
Brand#52	PROMO BURNISHED COPPER	49.00	4.00	Brand#53	LARGE BRUSHED NICKEL	23.00	4.00
Brand#52	PROMO BURNISHED NICKEL	9.00	4.00	Brand#53	LARGE BRUSHED NICKEL	36.00	4.00
Brand#52	PROMO BURNISHED STEEL	9.00	4.00	Brand#53	LARGE BRUSHED TIN	36.00	4.00
Brand#52	PROMO BURNISHED STEEL	23.00	4.00	Brand#53	LARGE BURNISHED BRASS	45.00	4.00
Brand#52	PROMO BURNISHED TIN	19.00	4.00	Brand#53	LARGE BURNISHED COPPER	19.00	4.00
Brand#52	PROMO BURNISHED TIN	36.00	4.00	Brand#53	LARGE BURNISHED COPPER	36.00	4.00
Brand#52	PROMO PLATED BRASS	19.00	4.00	Brand#53	LARGE BURNISHED NICKEL	23.00	4.00
Brand#52	PROMO PLATED BRASS	45.00	4.00	Brand#53	LARGE BURNISHED STEEL	19.00	4.00
Brand#52	PROMO PLATED BRASS	49.00	4.00	Brand#53	LARGE BURNISHED STEEL	23.00	4.00
Brand#52	PROMO PLATED COPPER	9.00	4.00	Brand#53	LARGE PLATED BRASS	9.00	4.00
Brand#52	PROMO PLATED NICKEL	3.00	4.00	Brand#53	LARGE PLATED BRASS	45.00	4.00
Brand#52	PROMO PLATED NICKEL	23.00	4.00	Brand#53	LARGE PLATED BRASS	49.00	4.00
Brand#52	PROMO POLISHED NICKEL	14.00	4.00	Brand#53	LARGE PLATED COPPER	23.00	4.00
Brand#52	PROMO POLISHED NICKEL	49.00	4.00	Brand#53	LARGE PLATED NICKEL	23.00	4.00
Brand#52	PROMO POLISHED TIN	36.00	4.00	Brand#53	LARGE PLATED NICKEL	49.00	4.00
Brand#52	SMALL ANODIZED BRASS	3.00	4.00	Brand#53	LARGE PLATED STEEL	49.00	4.00
Brand#52	SMALL ANODIZED BRASS	14.00	4.00	Brand#53	LARGE PLATED TIN	14.00	4.00
Brand#52	SMALL ANODIZED COPPER	3.00	4.00	Brand#53	LARGE POLISHED COPPER	49.00	4.00
Brand#52	SMALL ANODIZED NICKEL	36.00	4.00	Brand#53	LARGE POLISHED STEEL	36.00	4.00
Brand#52	SMALL ANODIZED STEEL	9.00	4.00	Brand#53	LARGE POLISHED TIN	9.00	4.00
Brand#52	SMALL ANODIZED STEEL	19.00	4.00	Brand#53	MEDIUM ANODIZED BRASS	23.00	4.00
Brand#52	SMALL BRUSHED NICKEL	19.00	4.00	Brand#53	MEDIUM ANODIZED STEEL	14.00	4.00
Brand#52	SMALL BRUSHED STEEL	23.00	4.00	Brand#53	MEDIUM ANODIZED STEEL	36.00	4.00
Brand#52	SMALL BRUSHED TIN	14.00	4.00	Brand#53	MEDIUM ANODIZED TIN	3.00	4.00
Brand#52	SMALL BRUSHED TIN	19.00	4.00	Brand#53	MEDIUM ANODIZED TIN	9.00	4.00
Brand#52	SMALL BURNISHED NICKEL	14.00	4.00	Brand#53	MEDIUM BRUSHED BRASS	3.00	4.00
Brand#52	SMALL BURNISHED NICKEL	49.00	4.00	Brand#53	MEDIUM BRUSHED COPPER	3.00	4.00
Brand#52	SMALL BURNISHED TIN	9.00	4.00	Brand#53	MEDIUM BRUSHED NICKEL	14.00	4.00
Brand#52	SMALL POLISHED BRASS	36.00	4.00	Brand#53	MEDIUM BRUSHED NICKEL	36.00	4.00
Brand#52	SMALL POLISHED BRASS	49.00	4.00	Brand#53	MEDIUM BRUSHED NICKEL	49.00	4.00
Brand#52	SMALL POLISHED TIN	45.00	4.00	Brand#53	MEDIUM BRUSHED STEEL	45.00	4.00
Brand#52	STANDARD ANODIZED BRASS	45.00	4.00	Brand#53	MEDIUM BURNISHED BRASS	3.00	4.00
Brand#52	STANDARD BRUSHED BRASS	23.00	4.00	Brand#53	MEDIUM BURNISHED BRASS	36.00	4.00
Brand#52	STANDARD BRUSHED COPPER	14.00	4.00	Brand#53	MEDIUM BURNISHED TIN	9.00	4.00
Brand#52	STANDARD BRUSHED TIN	36.00	4.00	Brand#53	MEDIUM BURNISHED TIN	14.00	4.00
Brand#52	STANDARD BURNISHED BRASS	49.00	4.00	Brand#53	MEDIUM BURNISHED TIN	36.00	4.00
Brand#52	STANDARD BURNISHED STEEL	19.00	4.00	Brand#53	MEDIUM PLATED BRASS	23.00	4.00
Brand#52	STANDARD BURNISHED TIN	9.00	4.00	Brand#53	MEDIUM PLATED COPPER	14.00	4.00

Compaq

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

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

```

Brand#55 STANDARD POLISHED STEEL 36.00 4.00
Brand#11 LARGE ANODIZED TIN 45.00 3.00
Brand#12 LARGE BURNISHED NICKEL 14.00 3.00
Brand#12 MEDIUM PLATED TIN 49.00 3.00
Brand#12 PROMO POLISHED TIN 3.00 3.00
Brand#13 STANDARD POLISHED NICKEL 19.00 3.00
Brand#15 SMALL BURNISHED STEEL 9.00 3.00
Brand#21 MEDIUM ANODIZED TIN 9.00 3.00
Brand#22 PROMO BRUSHED BRASS 19.00 3.00
Brand#22 PROMO BURNISHED COPPER 14.00 3.00
Brand#22 STANDARD BURNISHED COPPER 45.00 3.00
Brand#23 SMALL ANODIZED BRASS 23.00 3.00
Brand#24 PROMO BRUSHED STEEL 49.00 3.00
Brand#31 SMALL PLATED NICKEL 36.00 3.00
Brand#32 LARGE BRUSHED STEEL 45.00 3.00
Brand#41 PROMO BURNISHED NICKEL 36.00 3.00
Brand#43 STANDARD BRUSHED BRASS 23.00 3.00
Brand#44 MEDIUM ANODIZED NICKEL 9.00 3.00
Brand#51 MEDIUM BRUSHED TIN 36.00 3.00
Brand#53 MEDIUM BURNISHED BRASS 49.00 3.00
Brand#54 SMALL POLISHED BRASS 9.00 3.00

```

2762 rows processed.
Statement Processed in 10.06 seconds.

Ended Executing this Query at Mon Jun 15 12:59:06 1998

Query Started at 7613.47
Query Ended at 7623.53
Query Processed in 10.06 seconds.

-- Query 14 (Variant C)

Began Executing this Query at Mon Jun 15 12:59:06 1998

```

CREATE TABLE ALL_SALES0 (TYPE VARCHAR2(25), AMOUNT
NUMBER(20, 2))
tablespace TS_S

```

Statement Processed in 0.03 seconds.

```

CREATE TABLE SUM_PROMO_SALES0 (PROMO_AMOUNT
NUMBER(20, 2))
tablespace TS_S

```

Statement Processed in 0.03 seconds.

```

CREATE TABLE SUM_ALL_SALES0 (ALL_AMOUNT NUMBER(20,
2))
tablespace TS_S

```

Statement Processed in 0.03 seconds.

```

INSERT INTO ALL_SALES0
SELECT
P_TYPE,
SUM(L_EXTENDEDPRI * (1-L_DISCOUNT))
FROM LINEITEM, PARTS
WHERE
L_PARTKEY = P_PARTKEY
AND L_SHIPDATE >= TO_DATE('1995-09-01', 'YYYY-MM-DD')

```

```

AND L_SHIPDATE < ADD_MONTHS(TO_DATE('1995-09-01', 'YYYY-
MM-DD'), 1)
GROUP BY P_TYPE
Statement Processed in 1.08 seconds.

```

```

INSERT INTO SUM_PROMO_SALES0
SELECT
SUM(AMOUNT)
FROM ALL_SALES0
WHERE
TYPE LIKE 'PROMO%'
Statement Processed in 0.00 seconds.

```

```

INSERT INTO SUM_ALL_SALES0
SELECT
SUM(AMOUNT)
FROM ALL_SALES0
Statement Processed in 0.00 seconds.

```

```

SELECT
100.00 * PROMO_AMOUNT / ALL_AMOUNT AS PROMO_REVENUE
FROM SUM_PROMO_SALES0, SUM_ALL_SALES0

```

PROMO_REVENUE
16.73

1 row processed.
Statement Processed in 0.00 seconds.

DROP TABLE ALL_SALES0
Statement Processed in 0.04 seconds.

DROP TABLE SUM_PROMO_SALES0
Statement Processed in 0.04 seconds.

DROP TABLE SUM_ALL_SALES0
Statement Processed in 0.04 seconds.

Ended Executing this Query at Mon Jun 15 12:59:07 1998

Query Started at 7623.53
Query Ended at 7624.82
Query Processed in 1.29 seconds.

-- Query 8 (Variant B)

Began Executing this Query at Mon Jun 15 12:59:07 1998

```

SELECT
YEAR,
SUM(DECODE(NATION, 'BRAZIL', VOLUME, 0)) / SUM(VOLUME)
AS MKT_SHARE
FROM
(SELECT
TO_CHAR(O_ORDERDATE, 'YYYY') AS YEAR,
L_EXTENDEDPRI * (1-L_DISCOUNT) AS VOLUME,
N2.N_NAME AS NATION
FROM PARTS, SUPPLIER, LINEITEM, ORDERS, CUSTOMER,
NATION N1, NATION N2, REGION
WHERE
P_PARTKEY = L_PARTKEY
AND S_SUPPKEY = L_SUPPKEY

```

```

AND L_ORDERKEY = O_ORDERKEY
AND O_CUSTKEY = C_CUSTKEY
AND C_NATIONKEY = N1.N_NATIONKEY
AND N1.N_REGIONKEY = R_REGIONKEY
AND R_NAME = 'AMERICA'
AND S_NATIONKEY = N2.N_NATIONKEY
AND O_ORDERDATE BETWEEN TO_DATE('1995-01-01','YYYY-
MM-DD') AND TO_DATE('1996-12-31','YYYY-MM-DD')
AND P_TYPE = 'ECONOMY ANODIZED STEEL'
) ALL_NATIONS
GROUP BY YEAR
ORDER BY YEAR

```

```

YEAR MKT_SHARE
1995 0.05
1996 0.09

```

2 rows processed.
Statement Processed in 9.41 seconds.

Ended Executing this Query at Mon Jun 15 12:59:17 1998

Query Started at 7624.82
Query Ended at 7634.23
Query Processed in 9.41 seconds.

-- Query 12 (Variant B)

Began Executing this Query at Mon Jun 15 12:59:17 1998

```

SELECT
L_SHIPMODE,
SUM(DECODE(O_ORDERPRIORITY,'1-URGENT',1,'2-HIGH',1,0)) AS
HIGH_LINE_COUNT,
SUM(DECODE(O_ORDERPRIORITY,'1-URGENT',0,'2-HIGH',0,1)) AS
LOW_LINE_COUNT
FROM ORDERS, LINEITEM
WHERE
O_ORDERKEY = L_ORDERKEY
AND L_SHIPMODE IN ('MAIL','SHIP')
AND L_COMMITDATE < L_RECEIPTDATE
AND L_SHIPDATE < L_COMMITDATE
AND L_RECEIPTDATE >= TO_DATE('1994-01-01','YYYY-MM-DD')
AND L_RECEIPTDATE < ADD_MONTHS(TO_DATE('1994-01-01'
,'YYYY-MM-DD'),12)
GROUP BY L_SHIPMODE
ORDER BY L_SHIPMODE

```

```

L_SHIPMODE HIGH_LINE_COUNT LOW_LINE_COUNT
MAIL      654.00      950.00
SHIP      684.00     1004.00

```

2 rows processed.
Statement Processed in 0.92 seconds.

Ended Executing this Query at Mon Jun 15 12:59:18 1998

Query Started at 7634.23
Query Ended at 7635.16

Query Processed in 0.92 seconds.

-- Query 17 (Original)

Began Executing this Query at Mon Jun 15 12:59:18 1998

```

SELECT
SUM(L_EXTENDEDPRIE)/7.0 AS AVG_YEARLY
FROM LINEITEM, PARTS
WHERE
P_PARTKEY = L_PARTKEY
AND P_BRAND = 'Brand#23'
AND P_CONTAINER = 'MED BOX'
AND L_QUANTITY <
(SELECT
0.2*AVG(L_QUANTITY)
FROM LINEITEM
WHERE
L_PARTKEY = P_PARTKEY
)

```

```

AVG_YEARLY
24436.88

```

1 row processed.
Statement Processed in 0.62 seconds.
Setting the number of rows to fetch to: 10

Ended Executing this Query at Mon Jun 15 12:59:18 1998

Query Started at 7635.16
Query Ended at 7635.78
Query Processed in 0.62 seconds.

-- Query 3

Began Executing this Query at Mon Jun 15 12:59:18 1998

```

SELECT
L_ORDERKEY,
SUM(L_EXTENDEDPRIE*(1-L_DISCOUNT)) AS REVENUE,
TO_CHAR(O_ORDERDATE, 'YYYY-MM-DD'),
O_SHIPPRIORITY
FROM CUSTOMER, ORDERS, LINEITEM
WHERE
C_MKTSEGMENT = 'BUILDING'
AND C_CUSTKEY = O_CUSTKEY
AND L_ORDERKEY = O_ORDERKEY
AND O_ORDERDATE < TO_DATE('1995-03-15','YYYY-MM-DD')
AND L_SHIPDATE > TO_DATE('1995-03-15','YYYY-MM-DD')
GROUP BY L_ORDERKEY, O_ORDERDATE, O_SHIPPRIORITY
ORDER BY REVENUE DESC, O_ORDERDATE

```

```

L_ORDERKEY  REVENUE      TO_CHAR(O_ORDERD
O_SHIPPRIORITY
260930.00    320547.25    1995-03-12    0.00

```

402497.00	298879.53	1995-02-12	0.00
457859.00	296490.68	1995-01-17	0.00
509889.00	294068.87	1995-02-03	0.00
58117.00	292632.83	1995-02-21	0.00
538311.00	279666.00	1995-03-07	0.00
588421.00	275477.12	1995-03-03	0.00
416167.00	273765.45	1995-02-22	0.00
97830.00	273227.06	1995-03-04	0.00
90276.00	272233.92	1995-03-04	0.00

10 rows processed.
Statement Processed in 2.21 seconds.

Ended Executing this Query at Mon Jun 15 12:59:21 1998

Query Started at 7635.78
Query Ended at 7637.99
Query Processed in 2.21 seconds.

-- Query 5

Began Executing this Query at Mon Jun 15 12:59:21 1998

```

SELECT
N_NAME,
SUM(L_EXTENDEDPRI * (1-L_DISCOUNT)) AS REVENUE
FROM CUSTOMER, ORDERS, LINEITEM, SUPPLIER, NATION,
REGION
WHERE
C_CUSTKEY = O_CUSTKEY
AND O_ORDERKEY = L_ORDERKEY
AND L_SUPPKEY = S_SUPPKEY
AND C_NATIONKEY = S_NATIONKEY
AND S_NATIONKEY = N_NATIONKEY
AND N_REGIONKEY = R_REGIONKEY
AND R_NAME = 'ASIA'
AND O_ORDERDATE >= TO_DATE('1994-01-01','YYYY-MM-DD')
AND O_ORDERDATE < ADD_MONTHS(TO_DATE('1994-01-
01','YYYY-MM-DD'),12)
GROUP BY N_NAME
ORDER BY REVENUE DESC

```

N_NAME	REVENUE
CHINA	7349391.47
INDONESIA	6485853.40
INDIA	5505346.82
JAPAN	5388883.59
VIETNAM	4728846.60

5 rows processed.
Statement Processed in 7.21 seconds.

Ended Executing this Query at Mon Jun 15 12:59:28 1998

Query Started at 7637.99
Query Ended at 7645.20
Query Processed in 7.21 seconds.

-- Query 13 (Original)

Began Executing this Query at Mon Jun 15 12:59:28 1998

```

SELECT
YEAR,
SUM(REVENUE) AS REVENUE
FROM
(SELECT
TO_CHAR(O_ORDERDATE,'YYYY') AS YEAR,
L_EXTENDEDPRI * (1-L_DISCOUNT) AS REVENUE
FROM LINEITEM, ORDERS
WHERE
O_ORDERKEY = L_ORDERKEY
AND O_CLERK = 'Clerk#000000088'
AND L_RETURNFLAG = 'R'
) PERFORMANCE
GROUP BY YEAR
ORDER BY YEAR

```

YEAR REVENUE
1992 1262855.73
1993 964121.03
1994 1750395.29
1995 198820.30

4 rows processed.
Statement Processed in 2.56 seconds.

Ended Executing this Query at Mon Jun 15 12:59:30 1998

Query Started at 7645.20
Query Ended at 7647.77
Query Processed in 2.56 seconds.

-- Query 7 (Original)

Began Executing this Query at Mon Jun 15 12:59:30 1998

```

SELECT
SUPP_NATION,
CUST_NATION,
YEAR,
SUM(VOLUME) AS REVENUE
FROM
(SELECT
N1.N_NAME AS SUPP_NATION,
N2.N_NAME AS CUST_NATION,
TO_CHAR(L_SHIPDATE,'YYYY') AS YEAR,
L_EXTENDEDPRI * (1-L_DISCOUNT) AS VOLUME
FROM SUPPLIER, LINEITEM, ORDERS, CUSTOMER, NATION N1,
NATION N2
WHERE
S_SUPPKEY = L_SUPPKEY
AND O_ORDERKEY = L_ORDERKEY
AND C_CUSTKEY = O_CUSTKEY
AND S_NATIONKEY = N1.N_NATIONKEY
AND C_NATIONKEY = N2.N_NATIONKEY
AND ((N1.N_NAME = 'FRANCE' AND N2.N_NAME = 'GERMANY')
OR (N1.N_NAME = 'GERMANY' AND N2.N_NAME = 'FRANCE'))

```

AND L_SHIPDATE BETWEEN TO_DATE('1995-01-01','YYYY-MM-DD') AND TO_DATE('1996-12-31','YYYY-MM-DD')
) SHIPPING
 GROUP BY SUPP_NATION, CUST_NATION, YEAR
 ORDER BY SUPP_NATION, CUST_NATION, YEAR

SUPP_NATION	CUST_NATION	YEAR	REVENUE
FRANCE	GERMANY	1995	4611421.44
FRANCE	GERMANY	1996	4828420.37
GERMANY	FRANCE	1995	6755766.84
GERMANY	FRANCE	1996	5810951.40

4 rows processed.
 Statement Processed in 8.56 seconds.

Ended Executing this Query at Mon Jun 15 12:59:39 1998

Query Started at 7647.77
 Query Ended at 7656.33
 Query Processed in 8.56 seconds.

 -- Query 9 (Original)

Began Executing this Query at Mon Jun 15 12:59:39 1998

```

SELECT
NATION,
YEAR,
SUM(AMOUNT) AS SUM_PROFIT
FROM
(SELECT
N_NAME AS NATION,
TO_CHAR(O_ORDERDATE, 'YYYY') AS YEAR,
L_EXTENDEDPRI * (1-L_DISCOUNT) - PS_SUPPLYCOST *
L_QUANTITY AS AMOUNT
FROM PARTS, SUPPLIER, LINEITEM, PARTSUPP, ORDERS,
NATION
WHERE
S_SUPPKEY = L_SUPPKEY
AND PS_SUPPKEY = L_SUPPKEY
AND PS_PARTKEY = L_PARTKEY
AND P_PARTKEY = L_PARTKEY
AND O_ORDERKEY = L_ORDERKEY
AND S_NATIONKEY = N_NATIONKEY
AND P_NAME LIKE '%green%'
) PROFIT
GROUP BY NATION, YEAR
ORDER BY NATION, YEAR DESC

```

NATION	YEAR	SUM_PROFIT
ALGERIA	1998	1946316.01
ALGERIA	1997	2973825.69
ALGERIA	1996	3308881.52
ALGERIA	1995	3092227.30
ALGERIA	1994	3406958.71
ALGERIA	1993	3140744.03
ALGERIA	1992	3330704.41
ARGENTINA	1998	3045410.01
ARGENTINA	1997	4255378.59
ARGENTINA	1996	4651751.94
ARGENTINA	1995	4897797.00

ARGENTINA	1994	4823465.77
ARGENTINA	1993	4499810.71
ARGENTINA	1992	4764593.39
BRAZIL	1998	2932051.36
BRAZIL	1997	3784531.35
BRAZIL	1996	3965665.69
BRAZIL	1995	4063060.86
BRAZIL	1994	4236277.35
BRAZIL	1993	4363461.31
BRAZIL	1992	4684749.23
CANADA	1998	2217064.04
CANADA	1997	2950110.61
CANADA	1996	3184049.97
CANADA	1995	3962540.19
CANADA	1994	3365251.02
CANADA	1993	3617013.37
CANADA	1992	3407955.25
CHINA	1998	3048192.02
CHINA	1997	5001207.69
CHINA	1996	4800958.31
CHINA	1995	5154927.73
CHINA	1994	5882634.53
CHINA	1993	4733364.82
CHINA	1992	5014704.08
EGYPT	1998	1892538.74
EGYPT	1997	3849220.07
EGYPT	1996	3418656.55
EGYPT	1995	3766170.60
EGYPT	1994	3520025.56
EGYPT	1993	4375424.75
EGYPT	1992	4586034.39
ETHIOPIA	1998	1860117.73
ETHIOPIA	1997	3705722.33
ETHIOPIA	1996	3577215.39
ETHIOPIA	1995	3425219.55
ETHIOPIA	1994	3428616.18
ETHIOPIA	1993	3459815.43
ETHIOPIA	1992	3280072.91
FRANCE	1998	1592531.55
FRANCE	1997	2746176.54
FRANCE	1996	2505844.88
FRANCE	1995	2902077.00
FRANCE	1994	2532229.56
FRANCE	1993	2305725.44
FRANCE	1992	2955126.69
GERMANY	1998	3538625.73
GERMANY	1997	4425943.40
GERMANY	1996	4266344.96
GERMANY	1995	3952963.52
GERMANY	1994	4462655.80
GERMANY	1993	4435094.66
GERMANY	1992	4521715.41
INDIA	1998	3378369.34
INDIA	1997	4186477.85
INDIA	1996	5074383.92
INDIA	1995	4487435.38
INDIA	1994	4718312.63
INDIA	1993	4499573.81
INDIA	1992	4712930.33
INDONESIA	1998	2902077.10
INDONESIA	1997	4973644.23
INDONESIA	1996	4977652.49
INDONESIA	1995	5359380.15
INDONESIA	1994	4854637.20
INDONESIA	1993	4213131.42
INDONESIA	1992	4999478.51
IRAN	1998	2415763.10

IRAN 1997 4227175.11
 IRAN 1996 4527365.03
 IRAN 1995 4139514.72
 IRAN 1994 4166316.39
 IRAN 1993 3366959.59
 IRAN 1992 3599399.70
 IRAQ 1998 2596922.63
 IRAQ 1997 3707054.11
 IRAQ 1996 3726138.38
 IRAQ 1995 4350503.89
 IRAQ 1994 4131512.79
 IRAQ 1993 3787196.42
 IRAQ 1992 4043738.13
 JAPAN 1998 2265666.94
 JAPAN 1997 3988819.28
 JAPAN 1996 4319004.53
 JAPAN 1995 4262698.64
 JAPAN 1994 3545212.62
 JAPAN 1993 4051565.97
 JAPAN 1992 3692137.45
 JORDAN 1998 1978591.74
 JORDAN 1997 3315454.29
 JORDAN 1996 3236531.98
 JORDAN 1995 2778207.99
 JORDAN 1994 2420301.07
 JORDAN 1993 3272130.93
 JORDAN 1992 2649126.09
 KENYA 1998 2265677.73
 KENYA 1997 3493019.32
 KENYA 1996 3346373.30
 KENYA 1995 3537360.32
 KENYA 1994 2800950.72
 KENYA 1993 3477468.30
 KENYA 1992 2719618.04
 MOROCCO 1998 2549499.93
 MOROCCO 1997 3891824.90
 MOROCCO 1996 3730777.74
 MOROCCO 1995 3469641.13
 MOROCCO 1994 3747593.21
 MOROCCO 1993 3620742.70
 MOROCCO 1992 4303609.25
 MOZAMBIQUE 1998 2024719.46
 MOZAMBIQUE 1997 3706003.09
 MOZAMBIQUE 1996 3376430.93
 MOZAMBIQUE 1995 2737631.64
 MOZAMBIQUE 1994 3373146.48
 MOZAMBIQUE 1993 3608300.37
 MOZAMBIQUE 1992 3551263.95
 PERU 1998 2142791.97
 PERU 1997 4664076.15
 PERU 1996 3623628.93
 PERU 1995 3908939.79
 PERU 1994 3386204.16
 PERU 1993 3877048.49
 PERU 1992 3768394.25
 ROMANIA 1998 1760625.70
 ROMANIA 1997 2707685.33
 ROMANIA 1996 2553345.48
 ROMANIA 1995 2715901.59
 ROMANIA 1994 3023644.06
 ROMANIA 1993 2873247.32
 ROMANIA 1992 2728060.71
 RUSSIA 1998 2975973.22
 RUSSIA 1997 3785806.47
 RUSSIA 1996 4217625.59
 RUSSIA 1995 3883445.52
 RUSSIA 1994 4395855.01
 RUSSIA 1993 3900944.18

RUSSIA 1992 4691358.61
 SAUDI ARABIA 1998 2931482.83
 SAUDI ARABIA 1997 5498943.16
 SAUDI ARABIA 1996 4473723.74
 SAUDI ARABIA 1995 5939212.93
 SAUDI ARABIA 1994 4527695.71
 SAUDI ARABIA 1993 4928702.02
 SAUDI ARABIA 1992 5527261.52
 UNITED KINGDOM 1998 3198731.37
 UNITED KINGDOM 1997 4363882.74
 UNITED KINGDOM 1996 4730956.67
 UNITED KINGDOM 1995 4842014.55
 UNITED KINGDOM 1994 4912706.56
 UNITED KINGDOM 1993 4415255.96
 UNITED KINGDOM 1992 4375524.23
 UNITED STATES 1998 1892045.16
 UNITED STATES 1997 3102027.86
 UNITED STATES 1996 3334320.26
 UNITED STATES 1995 3168244.60
 UNITED STATES 1994 3296960.10
 UNITED STATES 1993 3558109.05
 UNITED STATES 1992 2755129.39
 VIETNAM 1998 2906627.03
 VIETNAM 1997 4544560.45
 VIETNAM 1996 4314259.00
 VIETNAM 1995 4365340.86
 VIETNAM 1994 3686987.71
 VIETNAM 1993 3764237.18
 VIETNAM 1992 3420922.00

175 rows processed.
 Statement Processed in 10.49 seconds.

Ended Executing this Query at Mon Jun 15 12:59:49 1998

Query Started at 7656.33
 Query Ended at 7666.81
 Query Processed in 10.49 seconds

SQL statements processed: 35
 Queries processed: 17

Appendix D: Seed and Query Substitution Parameters

Seed that was used is: 48480162

Query	Parameters				
1	110				
4	1997-01-01				
15	1997-01-01				
10	1994-09-01				
11	SAUDI ARABIA		0.0000010000		
6	1997-01-01	0.06	24		
2	42	BRASS	ASIA		
16	Brand#45	MEDIUM POLISHED	41	1	
	43	44	30	26	37
	10				
14	1997-03-01				
8	SAUDI ARABIA		MIDDLE EAST		LARGE
POLISHED BRASS					
12	FOB	MAIL	1995-01-01		
17	Brand#45	MED BOX			
3	MACHINERY		1995-03-20		
5	MIDDLE EAST		1996-01-01		
13	Clerk#000000825				
7	SAUDI ARABIA		MOROCCO		
9	sandy				

Appendix E: Implementation Specific Layer/Source Code

QEXECPL.C

```

/*=====
=====+
| Copyright (c) 1996 Oracle Corp, Redwood Shores, CA |
| OPEN SYSTEMS PERFORMANCE GROUP |
| All Rights Reserved |
=====+
=====+
| FILENAME |
| qexecpl.c |
| DESCRIPTION |
| SQL Execution Engine, OCI version |
| MODIFIED |
| pswong 04/02/96 - more polishing |
| pswong 03/25/96 - polish up |
| pswong 03/06/96 - created |
=====+
=====*/

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

#include "qexecpl.h"

/* Function Prototypes */

extern double gettime();

/* function prototypes from gen.c */

int get_statement();

/* Declare error handling functions */

void sql_error();
void sql_error2();

/* Other prototypes */

int define_output_variables();
void process_select_list();
void usage();
void SQLInit();
void SQLExec();
void SQLExit();
void *memalloc();

```

```

void print_header();
void print_rows();
int OFEN();
void remove_newline();

```

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

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

char stmt[SQL_LEN]; /* The SQL statement or comment line. */
char cmnt[81]; /* Buffer to save the comment. */

```

```

FILE *qtemp = stdin; /* fd for query template */
FILE *logfile = stdout; /* log and report files */
FILE *rep = stdout;

```

```

void *defbuf; /* Buffer pointer for ODEFIN */
int deflen = 0; /* Size of data type for ODEFIN */
int deftype = 1; /* Oracle type number for ODEFIN */

```

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

```

lddef tpclda;
csrdef curq;
unsigned long tpchda[256];

```

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

```
void usage() {
```

```

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

```

```

fprintf(stderr, "                (default is stdout)\n");
exit(-1);
}

void sql_error(lda, cur)
    ldadef *lda;
    csrdef *cur;
{
    char msg[2048];

    if (cur->rc) {
        oerhms(lda, cur->rc, (text *) msg, 2048);
        fprintf(stderr, "Error: SQL Error encountered in Oracle!\n");
        fputs(msg, stderr);
    }

    /* Rollback just in case */

    orol(lda);

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

    SQLexit();

    exit(1);
}

void sql_error2(lda, cur)
    ldadef *lda;
    csrdef *cur;
{
    char msg[2048];

    if (cur->rc) {
        oerhms(lda, cur->rc, (text *) msg, 2048);
        fprintf(stderr, "Error: SQL Error encountered in Oracle!\n");
        fputs(msg, stderr);
    }

    fflush(stderr);
}

void main(argc, argv)
    int argc;
    char *argv[];
{
    int retcode;    /* Return code for get_statement */

    /* Initialize some variables */

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

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

    strcpy(logname, argv[1]);

    /* Process optional parameters */

    argc -= 1;
    argv += 1;

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

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

    SQLinit();

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

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

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

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

        switch (retcode) {

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

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

```

```

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

/* Executes the query */

SQLExec();

s_tr_end = gettimeofday();
stm_cnt++;

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

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

/* Get Timing for the last query */

tr_end = gettimeofday();

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

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

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

fflush(rep);
fflush(logfile);

/* Close the query template file */

fclose(qtemp);

/* Disconnect from ORACLE. */

SQLExit();
exit(0);
}

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

void SQLInit() {
    int i;

    /* preallocate MAX_PREALLOC members of the dist array
    */

    /* initializes others to NULL so that we can determine who to free
    later */

    for (i=0; i<MAX_SEL_LIST; i++) {
        if (i < MAX_PREALLOC)
            dist[i] = (dtype *) memalloc (sizeof(dtype));
        else
            dist[i] = NULL;
    }

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

    if (orlon(&tpclda, (ub1 *)tpchda, (text *)logname, -1, (text *)0, -1, 0)) {
        /*
        if (olog(&tpclda, (ub1 *)tpchda, (text *)logname, -1, (text *)0, -1, (text
        *)0, -1,
            (ub4)OCI_LM_NBL)) {
        */
        fprintf(stderr, "Error: Failed to log on\n");
        sql_error(&tpclda, &tpclda);
        exit(-1);
    }

    printf("nConnected to ORACLE as user: %s\n\n", logname);

    /* Open a cursor for us to use */

    OOPEN(&tpclda, &curq);

}

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

void SQLExec()
{
    int i;

    if (!end_flag) {

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

        tr_end = gettimeofday();

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

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

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

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

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

```

```

fprintf(rep, "%s", cmnt);

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

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

fflush(logfile);
fflush(rep);

}

tr_start = tr_end;
qry_cnt++;

end_flag = 1;
}

s_tr_start = gettimeofday();

/* parse the query, use no defer option to determine the statement
type */
/* and catch any syntax error before we progress further. */

OPARSE(&tpclda, &curq, stmt, NA, FALSE, VER7);

/* Prints the query text to the logfile */

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

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

if (curq.ft != 4) {
    OEXEC(&tpclda &curq);
    return;
}

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

num_sel_list = define_output_variables();

/* Executes the query and fetches the rows */

(void) process_select_list(num_sel_list);

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

rows_ret = curq.rpc;

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

i=MAX_PREALLOC;

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

/* reset sel_fetchrows */

num_to_fetch = -1;

}

void SQLexit() {

    int i;

    oclose(&curq);
    ologof(&tpclda);

    /* free all memory */

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

    /* Flush all output */

    fflush(rep);
    fflush(logfile);

}

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

int define_output_variables()
{
    int i;

    for (i=0; i<MAX_SEL_LIST; i++) {
        slist[i].buflen = MAX_COLNAME_SIZE;

        if (odescr(&curq, POS(i), (sb4 *) &slist[i].dbsize,
            (sb2 *) &slist[i].dbtype, (sb1 *) &(slist[i].buff[0]),
            (sb4 *) &slist[i].buflen, (sb4 *) &slist[i].dszsize,
            (sb2 *) &slist[i].precision, (sb2 *) &slist[i].scale,
            (sb2 *) &slist[i].nullok)) {
            if (curq.rc == END_OF_LIST)
                break;
            else {
                sql_error(&tpclda, &curq);
                return -1;
            }
        }
    }

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

    if (slist[i].buflen < MAX_COLNAME_SIZE)
        (slist[i].buff[slist[i].buflen] = '\0');

    /* Well, we need to allocate for entries for dlist */

    if (i >= MAX_PREALLOC)
        dlist[i] = (dtype *) memalloc(sizeof(dtype));

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

```

```

switch (slist[i].dbtype) {

case NUM_TYPE:

    /* see if it is an integer or a floating point number */

    slist[i].dbsize = NUMWIDTH;

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

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

    break;

default:

    /* default is character string */

    defbuf = (char **) dist[i]->sbuf;
    deflen = MAX_STR_LEN;
    deftype = STR_TYPE;
    break;

}

/* Define the column */

ODEFIN(&tpclda, &curq, POS(i), defbuf, deflen, deftype, NA,
        (ub2 *)0, (text *)0, NA, NA, (ub2 *)dist[i]->rlen, (ub2 *)0);

}
return i;
}

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

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

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

    print_header(num);

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

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

    /* Fetch the rows and print them out */

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

        OEXFET(&tpclda, &curq, MAX_ARRAY, 0, 0);
        rows_ret = curq.rpc;
        print_rows(num, rows_ret);

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

        if (curq.rc != 1403) {
            if (num_to_fetch == -1) {
                while (OFEN(&tpclda, &curq, MAX_ARRAY) != -1) {
                    print_rows(num, (curq.rpc-rows_ret));
                    rows_ret = curq.rpc;
                }
                /* Print the final rows */
                print_rows(num, (curq.rpc-rows_ret));
                rows_ret = curq.rpc;
            } else {
                ntf -= MAX_ARRAY;
                while (OFEN(&tpclda, &curq, ((ntf > MAX_ARRAY) ?
MAX_ARRAY : ntf)) != -1) {
                    ntf -= MAX_ARRAY;
                    print_rows(num, (curq.rpc-rows_ret));
                    rows_ret = curq.rpc;
                    if (ntf <= 0) break;
                }
                print_rows(num, (curq.rpc-rows_ret));
                rows_ret = curq.rpc;
            }
        }
    } else {
        OEXFET(&tpclda, &curq, ntf, 0, 0);
        rows_ret = curq.rpc;
        print_rows(num, rows_ret);
    }

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

int OFEN(lda, cur, nrows)
    ldadef *lda;
    csrdef *cur;
    int nrows;
{
    if (ofen(cur, nrows))
        if (cur->rc != 1403) {
            sql_error(lda, cur);
            return 1;
        } else
            return -1;

    return 1;
}

int get_statement()
{
    char line[128];
    char *pos, *str;

```

```

/* Reset statement buffer */
stmt[0] = '\0';

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

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

    /* remove blanks */

    str = line;

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

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

    strcat(stmt, str);

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

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

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

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

void *memalloc(size)
    int size;
{
    void *tmp;

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

void print_header(nsel)
    int nsel; /* Number of select list items */
{
    int i;
    int len = 0; /* Running column length */
    int cwid = 0;

    fprintf(logfile, "\n");

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

        /* format the output a little */

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

        /* do a little bit of formatting */

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

#ifdef FORMAT1
    if ((slist[i].dbtype == INT_TYPE) || (slist[i].dbtype == FLT_TYPE))
        fprintf(logfile, "%*s", cwid, slist[i].buf);
    else /* string type */
        fprintf(logfile, "%*s", -cwid, slist[i].buf);
#else
    fprintf(logfile, "%*s", -cwid, slist[i].buf);
#endif /* FORMAT1 */
}

    fprintf(logfile, "\n");
}

void print_rows(ncol, nrow)
    int ncol;
    int nrow;
{
    int i, j;
    int len;
    int cwid;

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

        len = 0;

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

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

            /* do a little bit of formatting */

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

        switch(slist[j].dbtype) {
            case INT_TYPE:
#ifdef HAVE_SCALE
                fprintf(logfile, "%*ld", cwid, (dlist[j]->ibuf)[i]);
                break;

```

```
#endif /* HAVE_SCALE */
    case FLT_TYPE:
#ifdef FORMAT1
        fprintf(logfile, "%* 2f ", cwid, (dist[j]->fbuf)[i]);
#else
        fprintf(logfile, "%* 2f ", -cwid, (dist[j]->fbuf)[i]);
#endif /* FORMAT1 */
        break;
    default:
        fprintf(logfile, "%*s ", -cwid, (dist[j]->sbuf)[i]);
        break;
    }
}
fprintf(logfile, "\n");
}
}
```

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

```
void remove_newline(str)
    char *str;
{
    char *p;

    while ((p = strchr(str, '\n')) != NULL)
        *p = '\0';
}
```

QEXECPL.H

```
/*=====
=====+
|   Copyright (c) 1995 Oracle Corp. Redwood Shores, CA   |
|   OPEN SYSTEMS PERFORMANCE GROUP                       |
|   All Rights Reserved                                   |
=====+
=====+
| FILENAME
|   qstream.h
| DESCRIPTION
|   SQL statement execution front-end header file.
| MODIFIED
|   pswong 03/07/96 - created
=====+
=====*/

#ifndef QSTREAMPL_H

#define QSTREAMPL_H

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

#include <oratypes.h>
#include <ocidfn.h>

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

```
#endif /* __STDC__ */
```

```
typedef struct cda_def csrdef;
typedef struct cda_def ldadef;
```

/* some basic definitions */

```
#define UNAME_LEN 64
#define MAX_FILE_PATH_LEN 128
```

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

```
#ifndef FALSE
#define FALSE 1
#endif /* FALSE */
```

```
#define MAX(x,y) ((x >= y) ? x : y)
#define MIN(x,y) ((x <= y) ? x : y)
```

/* defines and typedefs for parsing */

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

/* defines and typedefs for query description */

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

```
#define END_OF_LIST 1007 /* Error code when we reach the end
of the */
```

/* select list. */

/* types for describe */

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

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

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

```
typedef struct des
{
    sb4 dbsize;
    sb2 dbtype;
    sb1 buf[MAX_COLNAME_SIZE];
    sb4 buflen;
    sb2 dsize;
    sb2 precision;
    sb2 scale;
    sb2 nullok;
} sltype;
```

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

```
#define MAX_ARRAY 3      /* Maximum array size for array fetch */
```

```
#define MAX_STR_LEN 256  /* Maximum size for string variables */
```

```
#define MAX_PREALLOC 8   /* Maximum number of preallocated select list */
```

```
/* definitions. */
```

```
/* Note: Well, I don't want to waste too much memory at the start. */  
/* Thus, I defined a MAX_PREALLOC so that we limit the amount */
```

```
/* of memory used for the dlist array. */  
/* The program will dynamically allocate the remaining members */
```

```
/* of the dlist array if necessary. */
```

```
/* Actually, I can go one step further by making everything */  
/* dynamic. But I don't want to call tons of mallocs and frees */  
/* during the query execution. Another one of those memory-CPU */  
/* trade-offs..... */
```

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

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

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

```
extern int errno;
```

```
#define SQL_LEN 2048
```

```
#ifndef NULL  
#define NULL 0  
#endif
```

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

```
#ifndef DISCARD  
# define DISCARD (void)  
#endif
```

```
#ifndef sword  
# define sword int  
#endif
```

```
#ifndef ub1  
#define ub1 unsigned char  
#endif
```

```
#define NA -1 /* ANSI SQL NULL */  
#define VER7 2
```

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

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

```
/* For get_statement */
```

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

```
#define OOPEN(lda, cursor)\  
    if (oopen((cursor), (lda), (text*)0, NA, NA, (text*)0, NA))\  
        {sql_error(lda, cursor);}\  
    else\  
        DISCARD 0
```

```
#define  
ODEFIN(lda, cursor, pos, buf, bufl, ftype, scale, indp, fmt, fmll, fmll, rlen, rcode  
e)\  
    if (odefin((cursor), (pos), (ub1*)(buf), (bufl), (ftype), (scale),\  
        (sb2*)(indp), (text*)(fmt), (fmll), (fmll), (rlen), (rcode)))\  
        {sql_error(lda, cursor);}\  
    else\  
        DISCARD 0
```

```
#define OEXFET(lda, cursor, nrow, cancel, exact)\  
    if (oexfet((cursor), (nrow), (cancel), (exact)))\  
        {if ((cursor)->rc == 1403) DISCARD 0; \  
        else {sql_error(lda, cursor);}} \  
    else\  
        DISCARD 0
```

```
#define OPARSE(lda, cursor, sqlstm, sql, defflg, lngflg)\  
    if (oparse((cursor), (text*)(sqlstm), (sb4)(sql), (defflg), (ub4)(lngflg)))\  
        {sql_error(lda, cursor);}\  
    else\  
        DISCARD 0
```

```
#define OEXEC(lda, cursor)\  
    if (oexec((cursor)))\  
        {sql_error(lda, cursor);}\  
    else\  
        DISCARD 0
```

```
#define ISOTXT "alter session set isolation_level = serializable"
```

```
#endif /* QSTREAMPL_H */
```

RUNPOWER1.SH

```
#!/bin/sh
```

```
#
```

```
=====
```

```
# Copyright (c) 1996 Oracle Corp, Redwood Shores, CA |  
# OPEN SYSTEMS PERFORMANCE DIVISION |  
# All Rights Reserved |  
#
```

```
=====
```

```
=====+
```

```
# FILENAME
```

```
# runpower1.sh
```

```
# DESCRIPTION
# Usage: runpower1.sh [-p <program for query stream>]
#           [-u1 <program for UF1>] [-u2 <program for UF2>]
#           [-o] [-s] [-h] [-u <user/password>]
#           <scale factor> <update parallelism>
#
# Single stream TPC-D Power Test.
#
=====
=====+

ORACLE_HOME=c:/orant
TPCD_HOME=c:/tpcd
SCRIPT_DIR=${TPCD_HOME}/scripts
SQL_DIR=${TPCD_HOME}/sql
UPD_DIR=${TPCD_HOME}/update
SRC_DIR=${TPCD_HOME}/source

RUN_ID_FILE=${TPCD_HOME}/r_id

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

TPCD_BIN=${TPCD_HOME}/bin
TPCD_LOG=${TPCD_HOME}/log
TPCD_RPT=${TPCD_HOME}/rpt

OUT=${TPCD_HOME}/out

GTIME=${SRC_DIR}/gtime

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

# The defaults

USER="tpcd/tpcd"
QPROG=${SRC_DIR}/qxec.exe

usage () {

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

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

if [ "$#" -ne "2" ]
then
usage
exit 1
fi

SF=$1
PARA=$2
THROUGHPUT=1

if [ $SF -eq 1 ]
then
QRY_DIR=${TPCD_HOME}/queries/q_one_0.sql
INIT_DIR=${TPCD_HOME}/dbs/onegig
elif [ $SF -eq 3 ]
then
QRY_DIR=${TPCD_HOME}/queries/q_three_0.sql
INIT_DIR=${TPCD_HOME}/dbs/three
elif [ $SF -eq 10 ]
then
QRY_DIR=${TPCD_HOME}/queries/q_ten_0.sql
INIT_DIR=${TPCD_HOME}/dbs/tengig
elif [ $SF -eq 100 ]
then
QRY_DIR=${TPCD_HOME}/queries/q_hundred_0.sql
INIT_DIR=${TPCD_HOME}/dbs/hundgig
elif [ $SF -eq 300 ]
then
QRY_DIR=${TPCD_HOME}/queries/q_threehundred_0.sql
INIT_DIR=${TPCD_HOME}/dbs/threehundred
elif [ $SF -eq 1000 ]
then
QRY_DIR=${TPCD_HOME}/queries/q_tbyte_0.sql
INIT_DIR=${TPCD_HOME}/dbs/tbyte
fi

export SF PARA THROUGHPUT

#${UPD_SPT}/genuf1.sh 1 ${PARA} ${SF}
#${UPD_SPT}/genuf1.sh -r 1 ${PARA} ${SF}
#${UPD_SPT}/genuf2.sh 1 ${PARA} ${SF}

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

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

echo "TPC-D Power Test Run `date`"
echo "RUNID is $RUN_ID"
echo ""

if [ $OSTAT ]
then
svrmgr80 @${ORACLE_HOME}/rdbms80/admin/utlbstat
fi
```

```

if [ $$STAT ]
then
    ${TPCD_BIN}/start_ntstat.sh $RUN_ID
fi

START=`$GTIME`
echo "TPC-D Power Test Execution starts at $START"
echo ""

# Execute UF1

echo "Start UF1 at `date`"

sh ${UPD_SPT}/runuf1.sh -u ${USER} ${RUN_ID} ${SF} 1 ${PARA} \
    > ${OUT}/uf1.${RUN_ID}.${HID} 2>&1

# Execute Query Stream

echo "End UF1. Start Query Stream at `date`"

${QPROG} ${USER} q${QRY_DIR}
r${TPCD_RPT}/rpt.${RUN_ID}.${HID} \
    !${OUT}/qs.${RUN_ID}.${HID} 2>&1

# Execute UF2

echo "End Query Stream. Start UF2 at `date`"

sh ${UPD_SPT}/runuf2.sh -u ${USER} ${RUN_ID} ${SF} 1 ${PARA} \
    > ${OUT}/uf2.${RUN_ID}.${HID} 2>&1

echo "END UF2 `date`"

END=`$GTIME`

echo "TPC-D Power Test Execution ends at $END"
echo "Measurement Interval is `echo $END - $START | bc`"
echo ""
echo "-----"
echo ""

if [ $OSTAT ]
then
    svrmgr30 @$ORACLE_HOME/rdbms80/admin/utlstat
    mv report.txt ora_stat_${RUN_ID}.txt
fi

if [ $$STAT ]
then
    ${TPCD_BIN}/kill_ntstat.sh ${RUN_ID}
    mv report_os.txt sys_stat_${RUN_ID}.txt
fi

echo "-- update function 1" >> ${TPCD_RPT}/rpt.${RUN_ID}.${HID}
awk -f utime.awk ${OUT}/uf1.${RUN_ID}.${HID} >>
    ${TPCD_RPT}/rpt.${RUN_ID}.${HID}
echo "-- update function 2" >> ${TPCD_RPT}/rpt.${RUN_ID}.${HID}
awk -f utime.awk ${OUT}/uf2.${RUN_ID}.${HID} >>
    ${TPCD_RPT}/rpt.${RUN_ID}.${HID}
cat ${TPCD_RPT}/rpt.${RUN_ID}.${HID} >
    ${TPCD_RPT}/pt_rpt.${RUN_ID}.${HID}
${TPCD_BIN}/metric ${SF} < ${TPCD_RPT}/rpt.${RUN_ID}.${HID} >>
    ${TPCD_RPT}/pt_rpt.${RUN_ID}.${HID}

#rm -f ${TPCD_RPT}/rpt.${RUN_ID}.${HID}
cat ${INIT_DIR}/new.ora >> ${TPCD_RPT}/rpt.${RUN_ID}.${HID}

```

```

echo "-- reset function 2 started at `date`"
sh ${UPD_SPT}/resuf2.sh -u ${USER} ${RUN_ID} ${SF} 1 ${PARA} \
    > ${OUT}/ruf2.${RUN_ID}.${HID} 2>&1
echo "-- reset function 1 started at `date`"
sh ${UPD_SPT}/resuf1.sh -u ${USER} ${RUN_ID} ${SF} 1 ${PARA} \
    > ${OUT}/ruf1.${RUN_ID}.${HID} 2>&1

```

RUNUF1.SH

```

#!/bin/sh
#
#=====
#
# Copyright (c) 1995 Oracle Corp, Redwood Shores, CA |
# OPEN SYSTEM PERFORMANCE GROUP |
# All Rights Reserved |
#=====
#
# FILENAME
# runuf1.sh
# DESCRIPTION
# runuf1.sh -l [<path name for reports>] -u [<uid/passwd>]
# -p [<program>] <run_id> <scale factor> <pair number>
# <parallelism>
# USAGE
# To execute UF1.
#=====

```

```

O=${TPCD_UPDATE_DATA}
TPCD_DIR=c:/tpcd
SCRIPT_DIR=${TPCD_DIR}/update/scripts
GTIME_DIR=${TPCD_DIR}/source
GTIME=${GTIME_DIR}/gtime
SQLLDR=c:/orant/bin/sqlldr80.exe
SQLPLUS=c:/orant/bin/plus80.exe
TS_S_DRVS=GHIJKLMNOPQRSTUVWXYZWXYEF
WXYEFABCDEFGHIJKLMNOPQRSTUVWXYZWXYEF
export TS_S_DRVS
TS_PS_DRVS=GHIJKLMNOPQRSTUVWXYZWXYEF
WXYEFABCDEFGHIJKLMNOPQRSTUVWXYZWXYEF
export TS_PS_DRVS

```

```

usage() {
echo ""
echo "runuf1.sh -l [<path name for reports>] -u [<uid/passwd>]"
echo "      -h <run_id> <scale factor> <pair number>"
echo "      <parallelism>"
echo ""
echo "run_id      : Run_ID of this update run."
echo "scale factor : Scale Factor of the database."
echo "set number  : The update pair number that this update function
belongs to."
echo "parallelism  : The parallelism of the updates."
echo ""
echo "-l      : Path name for reports on the update function. Debug
use for UF1 only"
echo "-u      : Userid/Password for Oracle. Default is tpcd/tpcd."
echo "-h      : To Display this message."
echo ""
}

```

```

LOGPATH=.
PASSWD="tpcd/tpcd"

```

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

```
while :
```

```
do
```

```
  case "$1" in
```

```
    -u) shift; PASSWD=$1;;
```

```
    -l) shift; LOGPATH=$1;;
```

```
    -h) usage; exit 0;;
```

```
    --) shift; break;;
```

```
  esac
```

```
  shift;
```

```
done
```

```
if [ $# -lt 4 ]
```

```
then
```

```
  usage
```

```
  exit 1
```

```
fi
```

```
RUN_ID=$1
```

```
SF=$2
```

```
SETNUM=$3
```

```
PAR=$4
```

```
if [ ${SF} -eq 0 ]
```

```
then
```

```
  UPDATE_DIR=${TPCD_DIR}/update/data/tenthgig
```

```
  LOG_DIR=${TPCD_DIR}/update/log/tenthgig
```

```
  DBF_DIR=\TPCD\DBS\TENTHGIG
```

```
fi
```

```
if [ ${SF} -eq 1 ]
```

```
then
```

```
  UPDATE_DIR=${TPCD_DIR}/update/data/onegig
```

```
  LOG_DIR=${TPCD_DIR}/update/log/onegig
```

```
  DBF_DIR=\TPCD\DBS\ONEGIG
```

```
fi
```

```
if [ ${SF} -eq 3 ]
```

```
then
```

```
  UPDATE_DIR=${TPCD_DIR}/update/data/threegig
```

```
  LOG_DIR=${TPCD_DIR}/update/log/threegig
```

```
  DBF_DIR=\TPCD\DBS\THREEGIG
```

```
fi
```

```
if [ ${SF} -eq 10 ]
```

```
then
```

```
  UPDATE_DIR=${TPCD_DIR}/update/data/tengig
```

```
  LOG_DIR=${TPCD_DIR}/update/log/tengig
```

```
  DBF_DIR=\TPCD\DBS\TENGIG
```

```
fi
```

```
if [ ${SF} -eq 30 ]
```

```
then
```

```
  UPDATE_DIR=${TPCD_DIR}/update/data/thirtygig
```

```
  LOG_DIR=${TPCD_DIR}/update/log/thirtygig
```

```
  DBF_DIR=\TPCD\DBS\THIRTYGIG
```

```
fi
```

```
if [ ${SF} -eq 100 ]
```

```
then
```

```
  UPDATE_DIR=${TPCD_DIR}/update/data/hundgig
```

```
  LOG_DIR=${TPCD_DIR}/update/log/hundgig
```

```
  DBF_DIR=\TPCD\DBS\HUNDGIG
```

```
fi
```

```
if [ ${SF} -eq 300 ]
```

```
then
```

```
  UPDATE_DIR=${TPCD_DIR}/update/data/threehundgig
```

```
  LOG_DIR=${TPCD_DIR}/update/log/threehundgig
```

```
  DBF_DIR=\TPCD\DBS\THREEHUNDGIG
```

```
fi
```

```
if [ ${SF} -eq 1000 ]
```

```
then
```

```
  UPDATE_DIR=${TPCD_DIR}/update/data/onetbyte
```

```
  LOG_DIR=${TPCD_DIR}/update/log/onetbyte
```

```
  DBF_DIR=\TPCD\DBS\ONETBYTE
```

```
fi
```

```
i=1
```

```
PID=""
```

```
# Let's figure out the extent sizes in Mb
```

```
LINEEXT=`echo $SF * 720 * 0.001 / ' $PAR + 1 | bc`
```

```
LINEEXT=`echo $LINEEXT | sed 's/\.[0123456789]*/s/'`
```

```
ORDEXT=`echo $SF * 200 * 0.001 / ' $PAR + 1 | bc`
```

```
ORDEXT=`echo $ORDEXT | sed 's/\.[0123456789]*/s/'`
```

```
# perform the update function 1
```

```
START=`SGTIME`
```

```
# first create the temp tables
```

```
SQLPLUS $PASSWD <<!
```

```
drop table temp_item;
```

```
create table temp_item (
```

```
  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      varchar(10) ,
```

```
  l_linenumber    number ,
```

```
  l_shipinstruct  varchar(25) ,
```

```
  l_comment       varchar(44)
```

```
)
```

```
pctfree 1
```

```
pctused 99
```

```
tablespace ts_s
```

```
storage (initial 1m next ${LINEEXT}m pctincrease 0)
```

```
parallel (degree 40 instances 1)
```

```
nologging
```

```
;
```

```
drop table temp_ord;
```

```
create table temp_ord (
```

```
  o_orderdate     date ,
```

```
  o_orderkey      number ,
```

```
  o_custkey       number ,
```

```
  o_orderpriority varchar(15) ,
```

```
  o_shippriority  number ,
```

```
  o_clerk         varchar(15) ,
```

```
  o_orderstatus   char(1) ,
```


[illegible]

```

(select /*+ PARALLEL(temp_okey,1) */ * from temp_okey);

delete /*+ PARALLEL(orders,40) INDEX(ORDERS, O_OP) */
from orders where o_orderkey in
(select /*+ PARALLEL(temp_okey,1) */ * from temp_okey);

commit;

drop table temp_okey;
spool off;
exit;
!

END=$GTIME

# Done

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

RESUF1.SH
#!/bin/sh
#
#=====
#
# Copyright (c) 1995 Oracle Corp. Redwood Shores, CA |
# OPEN SYSTEM PERFORMANCE GROUP |
# All Rights Reserved |
#=====
#
# FILENAME
# resuf1.sh
# DESCRIPTION
# resuf1.sh [-u <uid/passwd to login>] <run_id>
# <scale factor> <pair number> <parallelism>
# USAGE
# To restore UF1.
#=====
#
O=${TPCD_UPDATE_DATA}
TPCD_DIR=c:/tpcd
SCRIPT_DIR=${TPCD_DIR}/update/scripts
GTIME_DIR=${TPCD_DIR}/source
GTIME=${GTIME_DIR}/gtime
TS_S_DRVS=GHIJKLMNOPQRSTUVWXYZWXYFGHIJKLMNOPQRSTUVWXYZ
WXYFGHIJKLMNOPQRSTUVWXYZWXYFGHIJKLMNOPQRSTUVWXYZ
TS_PS_DRVS=GHIJKLMNOPQRSTUVWXYZWXYFGHIJKLMNOPQRSTUVWXYZ
WXYFGHIJKLMNOPQRSTUVWXYZWXYFGHIJKLMNOPQRSTUVWXYZ
SQLPLUS=c:/orant/bin/plus80.exe
SQLLDR=c:/orant/bin/sqlldr80.exe

usage() {
echo ""
echo "resuf1.sh [-u <user/passwd>] <run_id> <scale factor> <pair
number> <parallelism>"
echo ""
echo "run_id : Run_ID of this update run."
echo "scale factor : Scale Factor of the database."
echo "set number : The update pair number that this update function
belongs to."
echo "parallelism : The parallelism of the updates."
echo ""
echo "-u : Userid/Password for Oracle. Default is tpcd/tpcd"
echo "-h : To Display this message."

```

```

echo ""
}

PASSWD="tpcd/tpcd"

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

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

if [ $# -lt 4 ]
then
usage
exit 1
fi

RUN_ID=$1
SF=$2
SETNUM=$3
PAR=$4

if [ ${SF} -eq 0 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/tenthgig
LOG_DIR=${TPCD_DIR}/update/log/tenthgig
DBF_DIR=\\TPCD\\DBS\\TENTHGIG
fi

if [ ${SF} -eq 1 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/onegig
LOG_DIR=${TPCD_DIR}/update/log/onegig
DBF_DIR=\\TPCD\\DBS\\ONEGIG
fi

if [ ${SF} -eq 3 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/threegig
LOG_DIR=${TPCD_DIR}/update/log/threegig
DBF_DIR=\\TPCD\\DBS\\THREEGIG
fi

if [ ${SF} -eq 10 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/tengig
LOG_DIR=${TPCD_DIR}/update/log/tengig
DBF_DIR=\\TPCD\\DBS\\TENGIG
fi

if [ ${SF} -eq 30 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/thirtygig
LOG_DIR=${TPCD_DIR}/update/log/thirtygig
DBF_DIR=\\TPCD\\DBS\\THIRTYGIG
fi

if [ ${SF} -eq 100 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/hundgig

```

```
LOG_DIR=${TPCD_DIR}/update/log/hundgig
DBF_DIR=\TPCD\DBS\HUNDGIG
fi

if [ ${SF} -eq 300 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/threehundgig
LOG_DIR=${TPCD_DIR}/update/log/threehundgig
DBF_DIR=\TPCD\DBS\THREEHUNDGIG
fi

if [ ${SF} -eq 1000 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/onetbyte
LOG_DIR=${TPCD_DIR}/update/log/onetbyte
DBF_DIR=\TPCD\DBS\ONETBYTE
fi

i=1
PID=""

START=`$GTIME`

# first create the temp tables

$SQLPLUS $PASSWD <<|

drop table temp_okey;
create table temp_okey (
    t_orderkey      number
)
pctfree 1
pctused 99
tablespace ts_ps
storage (initial 64k next 1m pctincrease 0)
parallel (degree 60 instances 1)
nologging
;

exit;
!

# All processes have started, now wait for all the update processes to
finish.

wait

while [ $i -le $PAR ]
do
    FN=`expr $i - 1`
    FN=`expr $FN % 40`
    FN=`expr $FN + 1`

    $SQLLDR userid=$PASSWD control=${SCRIPT_DIR}/tempokey.ctl \
log=${LOG_DIR}/ti${i}_ruf1.log \
data=${UPDATE_DIR}/okey.u${SETNUM}.${i} \
direct=true parallel=true \
FILE=\VIII\..\\MORA_PS_${FN} &

    i=`expr $i + 1`
done

wait

$SQLPLUS $PASSWD <<|

spool resuf1.log;
```

```
set timing on;
analyze table temp_okey estimate statistics sample 1000 rows;
alter session force parallel dml;
```

```
delete /*+ PARALLEL(lineitem,60) INDEX(lineitem,l_ored) */
from lineitem where l_orderkey in
(select /*+ PARALLEL(temp_okey,1) */
 * from temp_okey);
```

```
delete /*+ PARALLEL(orders,60) INDEX(orders,o_op) */
from orders where o_orderkey in
(select /*+ PARALLEL(temp_okey,1) */
 * from temp_okey);
```

```
commit;
```

```
drop table temp_okey;
spool off;
exit;
!
```

```
END=`$GTIME`
```

```
# Done
```

```
echo ""
echo "Update Function 2 (resuf1) Set $SETNUM done!"
echo "Elapsed Time is `echo $END - $START | bc`"
echo ""
```

RESUF2.SH

```
#!/bin/sh
#
#=====
#      Copyright (c) 1995 Oracle Corp, Redwood Shores, CA      |
#      OPEN SYSTEM PERFORMANCE GROUP                            |
#      All Rights Reserved                                       |
#=====
# FILENAME
#   resuf2.sh
# DESCRIPTION
#   resuf2.sh -l [<path name for reports>] -u [<uid/passwd>]
#           <run_id> <scale factor> <pair number>
#           <parallelism>
# USAGE
#   To execute UF1.
#=====
O=${TPCD_UPDATE_DATA}
TPCD_DIR=c:/tpcd
SCRIPT_DIR=${TPCD_DIR}/update/scripts
GTIME_DIR=${TPCD_DIR}/source
GTIME=${GTIME_DIR}/gtime
SQLPLUS=c:/orant/bin/plus80.exe
SQLLDR=c:/orant/bin/sqlldr80.exe
TS_S_DRVS=GHIJKLMNOPQRSTUVWXYZVWXYEFGHIJKLMNOPQRSTUVWXYZ
WXYEFGHIJKLMNOPQRSTUVWXYZVWXYEF
TS_PS_DRVS=GHIJKLMNOPQRSTUVWXYZVWXYEFGHIJKLMNOPQRSTUVWXYZ
WXYEFGHIJKLMNOPQRSTUVWXYZVWXYEF

usage() {
echo ""
echo "resuf2.sh -l [<path name for reports>] -u [<uid/passwd>]"
echo "      -h <run_id> <scale factor> <pair number>"
echo "      <parallelism>"
}
```

```

echo ""
echo "run_id      : Run_ID of this update run."
echo "scale factor : Scale Factor of the database."
echo "set number   : The update pair number that this update function
belongs to."
echo "parallelism  : The parallelism of the updates."
echo ""
echo "-l          : Path name for reports on the update function.  Debug
use for UF1 only"
echo "-u          : Userid/Password for Oracle. Default is tpcd/tpcd."
echo "-h          : To Display this message."
echo ""
}

```

```

LOGPATH=
PASSWD="tpcd/tpcd"

```

```

set -- 'getopt "l:u:h" "$@"' || usage

```

```

while :
do
    case "$1" in
        -u) shift; PASSWD=$1;;
        -l) shift; LOGPATH=$1;;
        -h) usage; exit 0;;
        --) shift; break;;
        esac
    shift;
done

```

```

if [ $# -lt 4 ]
then
    usage
    exit 1
fi

```

```

RUN_ID=$1
SF=$2
SETNUM=$3
PAR=$4

```

```

if [ ${SF} -eq 0 ]
then
    UPDATE_DIR=${TPCD_DIR}/update/data/tenthgig
    LOG_DIR=${TPCD_DIR}/update/log/tenthgig
    DBF_DIR=\TPCD\DBS\TENTHGIG
fi

```

```

if [ ${SF} -eq 1 ]
then
    UPDATE_DIR=${TPCD_DIR}/update/data/onegig
    LOG_DIR=${TPCD_DIR}/update/log/onegig
    DBF_DIR=\TPCD\DBS\ONEGIG
fi

```

```

if [ ${SF} -eq 3 ]
then
    UPDATE_DIR=${TPCD_DIR}/update/data/threegig
    LOG_DIR=${TPCD_DIR}/update/log/threegig
    DBF_DIR=\TPCD\DBS\THREEGIG
fi

```

```

if [ ${SF} -eq 10 ]
then
    UPDATE_DIR=${TPCD_DIR}/update/data/tengig
    LOG_DIR=${TPCD_DIR}/update/log/tengig

```

```

    DBF_DIR=\TPCD\DBS\TENGIG
fi

```

```

if [ ${SF} -eq 30 ]
then
    UPDATE_DIR=${TPCD_DIR}/update/data/thirtygig
    LOG_DIR=${TPCD_DIR}/update/log/thirtygig
    DBF_DIR=\TPCD\DBS\THIRTYGIG
fi

```

```

if [ ${SF} -eq 100 ]
then
    UPDATE_DIR=${TPCD_DIR}/update/data/hundgig
    LOG_DIR=${TPCD_DIR}/update/log/hundgig
    DBF_DIR=\TPCD\DBS\HUNDGIG
fi

```

```

if [ ${SF} -eq 300 ]
then
    UPDATE_DIR=${TPCD_DIR}/update/data/threehundgig
    LOG_DIR=${TPCD_DIR}/update/log/threehundgig
    DBF_DIR=\TPCD\DBS\THREEHUNDGIG
fi

```

```

if [ ${SF} -eq 1000 ]
then
    UPDATE_DIR=${TPCD_DIR}/update/data/onetbyte
    LOG_DIR=${TPCD_DIR}/update/log/onetbyte
    DBF_DIR=\TPCD\DBS\ONETBYTE
fi

```

```

i=1
PID=""

```

```

# Let's figure out the extent sizes in Mb
LINEEXT='echo $SF * 720 * 0.001 / ' $PAR + 1 | bc'
LINEEXT='echo $LINEEXT | sed 'A.[0123456789]*/s///'
ORDEXT='echo $SF * 200 * 0.001 / ' $PAR + 1 | bc'
ORDEXT='echo $ORDEXT | sed 'A.[0123456789]*/s///'

```

```

# perform the update function 1

```

```

START=`$GTIME`

```

```

# first create the temp tables

```

```

$SQLPLUS $PASSWD <<|

```

```

drop table temp_item;
create table temp_item (
    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       varchar(10) ,
    l_linenum        number ,
    l_shipinstruct   varchar(25) ,
    l_comment        varchar(44)

```

```

)
pctfree 1
pctused 99
tablespace ts_ps
storage (initial 64k next ${LINEEXT}m pctincrease 0)
parallel (degree 60 instances 1)
nologging
;

drop table temp_ord;
create table temp_ord (
    o_orderdate      date,
    o_orderkey        number,
    o_custkey         number,
    o_orderpriority   varchar(15),
    o_shippriority    number,
    o_clerk           varchar(15),
    o_orderstatus     char(1),
    o_totalprice      number,
    o_comment         varchar(79)
)
pctfree 1
pctused 99
tablespace ts_ps
storage (initial 64k next ${ORDEXT}m pctincrease 0)
parallel (degree 60 instances 1)
nologging
;

exit;
!

while [ $i -le $PAR ]
do
    FN='expr $i - 1'
    FN='expr $FN % 40'
    FN='expr $FN + 1'

    # Kick off sqlldr to load the data into the temporary staging tables.

    $SQLLDR userid=$PASSWD control=${SCRIPT_DIR}/tempitem.ctl \
    log=${LOG_DIR}/ti${i}_ruf1.log \
    data=${UPDATE_DIR}/rlineitem.tbl.u${SETNUM}. ${i} \
    direct=true parallel=true \
    FILE=\\|||||\\|||ORA_PS_${FN} &

    $SQLLDR userid=$PASSWD control=${SCRIPT_DIR}/tempord.ctl \
    log=${LOG_DIR}/to${i}_ruf2.log \
    data=${UPDATE_DIR}/rorder.tbl.u${SETNUM}. ${i} \
    direct=true parallel=true \
    FILE=\\|||||\\|||ORA_PS_${FN} &

    i='expr $i + 1'

done

# All sqlldr processes have started, now wait for all them to finish.

wait

# now do the insert and then drop the staging tables

$SQLPLUS $PASSWD <<|

spool resuf2.log
set timing on;

analyze table temp_ord estimate statistics sample 1000 rows;

```

```

analyze table temp_item estimate statistics sample 1000 rows;

alter session force parallel dml;

insert /*+ PARALLEL(lineitem, 60) */
into lineitem
(select /*+ PARALLEL(temp_item, 1) */ * from temp_item);

insert /*+ PARALLEL(orders,60) */
into orders
(select /*+ PARALLEL(temp_ord, 1) */ * from temp_ord);

rem rollback for now
rem rollback;

commit;

drop table temp_item;
drop table temp_ord;
spool off;
exit;
!

END= '$GTIME'

# Done

echo ""
echo "Update Function 1 (resuf2) Set $SETNUM done!"
echo "Elapsed Time is `echo $END - $START | bc`"
echo ""

```

ATRANSPL.C

```

/*=====
=====+
|      Copyright (c) 1995 Oracle Corp. Redwood Shores, CA      |
|      OPEN SYSTEMS PERFORMANCE GROUP                          |
|      All Rights Reserved                                      |
=====+
=====+
| FILENAME
|   atranspl.c
| DESCRIPTION
|   TPC-D benchmark ACID transaction driver, OCI version
| MODIFIED
|   pswong 01/06/96 - created
|   jfee   02/09/97 - ported to NT
=====+
=====*/
#include <windows.h>

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

/* Declare error handling functions */

double gettime();
void sql_error();
void usage();
void ACIDinit();
int atoi();
int getpid();

```

```

/*
Note: on NT, drand48, srand48 and lrand48 are changed to
      rand, srand and rand respectively.
int drand48()
void srand48();
long lrand48();
*/

void srand();
int rand();

/* 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;
double l_quan = 0.0;
double l_newquan = 0.0;
double l_eprice = 0.0;
double l_neweprice = 0.0;
double l_disc = 0.0;
double l_tax = 0.0;

/* other declarations */

float delta = 0.0F;
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 */

FILE *logfile = NULL; /* logfile for durability (optional) */
FILE *infile = stdin; /* input file (optional) */
/* in the format of <o_key> <delta> */
FILE *outfile = stdout; /* output file (optional) */

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

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 */

ldadef tpclda;
csrdef curi, curr, cure1, cure2;

unsigned long tpchda[256];

```

```

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[<pathname for input>] [<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/tcpd\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 sql_error(lda, cur)
ldadef *lda;
csrdef *cur;
{
    text msg[2048];

    if (cur->rc) {
        oerhms(lda, cur->rc, msg, 2048);
        fprintf(stderr, "Error: SQL Error encountered in Oracle\n");
        fprintf(stderr, "%s\n", msg);
    }

    orol(lda);
    ologof(lda);

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

    exit(-1);
}

void main(argc, argv)
int argc;
char *argv[];
{

```

```
// int i;
char line[64];

/* Initialize some variables */

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

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

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

proc_no = atoi(argv[1]);

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

num_streams = atoi(argv[2]);

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

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

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

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

/* Process optional parameters */

argc -= 4;
argv += 4;

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

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

/* Initialize the cursors etc. */

(void) ACIDinit();

/* sleep for some time (triggering) */

Sleep(trig*1000);

/* start doing the ACID transactions */

tr_start = gettimeofday();

/* 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 %f\n", &o_key, &delta);

    /* Obtain l_key from l_key query */

    OEXEC(&tpclda, &curi);

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

    l_key = (lrand() % l_key) + 1;
    l_key = (rand() % l_key) + 1;

#else
    sscanf(line, "%d %d %f\n", &o_key, &l_key, &delta);
#endif /* NOLKEY */

    /* Generate delta if necessary */

    if (BIT(flag, DELTA))
        /* delta = floor((drand48() * 100)) + 1.0; */
        delta = (float) (rand() % 100 + 1);

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

    curr_time = time(NULL);

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

    fprintf(outfile, "o_key: %ld\n", (long) o_key);
    fprintf(outfile, "l_key: %ld\n", (long) l_key);
}
```

```

fprintf(outfile, "delta: %.2f\n", (float) delta);
fflush(outfile);

OEXEC(&tpclda, &curr);

curr_time = time(NULL);

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

fprintf(outfile, "Sleep %d seconds before
COMMIT/ROLLBACK...\n\n", slp);
Sleep(slp*1000);

/* Shall we commit? */

if (BIT(flag, COMMIT)) {
    OCOM(&tpclda, &tpclda);
    curr_time = time(NULL);
    fprintf(outfile, "ACID Transaction iteration %d COMMITED at
%s\n",
            num_iter, ctime(&curr_time));
} else {
    OROL(&tpclda, &tpclda);
    curr_time = time(NULL);
    fprintf(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. */

fflush(outfile);

/* Report initial and new values for o_totalprice, l_extendedprice, */
/* l_quantity. */
/*
curr_time = time(NULL);
fprintf(outfile, "Transaction Completed at %s\n", ctime(&curr_time));
*/
/* Get the values in LINEITEM and ORDERS after the transaction */

if (BIT(flag, LOGFILE)) {
    fprintf(logfile, "p_key:    %ld\n", (long) l_pkey);
    fprintf(logfile, "s_key:    %ld\n", (long) l_skey);
    fprintf(logfile, "o_key:    %ld\n", (long) o_key);
    fprintf(logfile, "l_key:    %ld\n", (long) l_key);
    fprintf(logfile, "delta:    %.2f\n", (float) delta);
    fprintf(logfile, "Transaction Completed at %s\n",
ctime(&curr_time));
    fprintf(logfile, "-----\n");
    fflush(logfile);
} else {
    OEXEC(&tpclda, &cure1);
    OEXEC(&tpclda, &cure2);
    fprintf(outfile, "AFTER TRANSACTION:\n");
    fprintf(outfile, "l_extendedprice: %.2f\n", l_neweprice);
    fprintf(outfile, "l_quantity:    %.2f\n", l_newquan);
    fprintf(outfile, "o_totalprice:  %.2f\n\n", o_newtprice);
    fprintf(outfile, "l_tax:        %.2f\n", l_tax);
    fprintf(outfile, "l_discount:   %.2f\n", l_disc);
    fprintf(outfile, "rprice:      %.2f\n", rprice);
    fprintf(outfile, "cost:        %.2f\n", cost);

```

```

        fprintf(outfile, "-----\n");
    }
}

tr_end = gettimeofday();

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

/* Disconnect from ORACLE. */

fflush(outfile);

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

oclose(&curei);
oclose(&curr);
oclose(&cure1);
oclose(&cure2);

exit(0);
}

void ACIDinit()
{
    /* run random seed */

    /* srand48(getpid()); */
    srand(getpid());

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

    if (orlon(&tpclda, (ub1 *) tpclda, (text *) lname, -1, (text *) 0, -1, 0)) {
        fprintf(stderr, "Error in ACID process %d: Failed to log on\n",
proc_no);
        sql_error(&tpclda, &tpclda);
        exit(-1);
    }

    /* Disables auto commit */

    if (ocof(&tpclda)) {
        sql_error(&tpclda, &tpclda);
        ologoff(&tpclda);
        exit(-1);
    }
}

```

```

/* Make session serializable */

OOPEN(&tpclda, &curi);
sprintf((char *) sqlstmt, ISOTXT);
OPARSE(&tpclda, &curi, sqlstmt, NA, FALSE, VER7);
OEXEC(&tpclda, &curi);

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. */

OOPEN(&tpclda, &curi);
sprintf((char *) sqlstmt, SQLTXT1);
OPARSE(&tpclda, &curi, sqlstmt, NA, FALSE, VER7);

OBNDRV(&tpclda, &curi, ":l_key", ADR(l_key), SIZ(l_key), SQLT_INT);

OBNDRV(&tpclda, &curi, ":o_key", ADR(o_key), SIZ(o_key), SQLT_INT);

#endif /* NOLKEY */

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

OOPEN(&tpclda, &curr);
sprintf((char *) sqlstmt, SQLTXT2);
OPARSE(&tpclda, &curr, (text *)sqlstmt, NA, FALSE, VER7);

/* bind variables */

OBNDRV(&tpclda, &curr, ":l_key", ADR(l_key), SIZ(l_key), SQLT_INT);
OBNDRV(&tpclda,
&curr, ":o_key", ADR(o_key), SIZ(o_key), SQLT_INT);
OBNDRV(&tpclda, &curr, ":delta", ADR(delta), SIZ(delta), SQLT_FLT);
OBNDRV(&tpclda,
&curr, ":l_pkey", ADR(l_pkey), SIZ(l_pkey), SQLT_INT);
OBNDRV(&tpclda,
&curr, ":l_skey", ADR(l_skey), SIZ(l_skey), SQLT_INT);
OBNDRV(&tpclda,
&curr, ":l_qun", ADR(l_qun), SIZ(l_qun), SQLT_FLT);
OBNDRV(&tpclda,
&curr, ":l_newqun", ADR(l_newqun), SIZ(l_newqun), SQLT_FLT);
OBNDRV(&tpclda, &curr, ":l_tax", ADR(l_tax), SIZ(l_tax), SQLT_FLT);
OBNDRV(&tpclda,
&curr, ":l_disc", ADR(l_disc), SIZ(l_disc), SQLT_FLT);
OBNDRV(&tpclda,
&curr, ":l_eprice", ADR(l_eprice), SIZ(l_eprice), SQLT_FLT);
OBNDRV(&tpclda,
&curr, ":l_neweprice", ADR(l_neweprice), SIZ(l_neweprice), SQLT_FLT);
OBNDRV(&tpclda,
&curr, ":o_tprice", ADR(o_tprice), SIZ(o_tprice), SQLT_FLT);
OBNDRV(&tpclda,
&curr, ":o_newtprice", ADR(o_newtprice), SIZ(o_newtprice), SQLT_FLT);
OBNDRV(&tpclda,
&curr, ":rprice", ADR(rprice), SIZ(rprice), SQLT_FLT);
OBNDRV(&tpclda, &curr, ":cost", ADR(cost), SIZ(cost), SQLT_FLT);

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

OOPEN(&tpclda, &cure1);
sprintf((char *) sqlstmt, SQLTXT3);
OPARSE(&tpclda, &cure1, (text *)sqlstmt, NA, FALSE, VER7);

OOPEN(&tpclda, &cure2);

```

```

sprintf((char *) sqlstmt, SQLTXT4);
OPARSE(&tpclda, &cure2, (text *)sqlstmt, NA, FALSE, VER7);

```

```

/* bind variables */

OBNDRV(&tpclda,
&cure1, ":l_neweprice", ADR(l_neweprice), SIZ(l_neweprice),
SQLT_FLT);
OBNDRV(&tpclda,
&cure1, ":l_newqun", ADR(l_newqun), SIZ(l_newqun), SQLT_FLT);
OBNDRV(&tpclda,
&cure1, ":o_key", ADR(o_key), SIZ(o_key), SQLT_INT);
OBNDRV(&tpclda,
&cure1, ":l_key", ADR(l_key), SIZ(l_key), SQLT_INT);

OBNDRV(&tpclda,
&cure2, ":o_newtprice", ADR(o_newtprice), SIZ(o_newtprice),
SQLT_FLT);
OBNDRV(&tpclda,
&cure2, ":o_key", ADR(o_key), SIZ(o_key), SQLT_INT);
}

```

ATRANSPL.H

```

/*=====
=====+
| Copyright (c) 1995 Oracle Corp. Redwood Shores, CA |
| OPEN SYSTEMS PERFORMANCE GROUP |
| All Rights Reserved |
=====+
=====+
| FILENAME
| atrans.h
| DESCRIPTION
| TPC-D benchmark ACID transaction driver header, OCI version
| MODIFIED
| pswong 01/06/96 - created
| fjee 02/10/97 - Ported to NT
=====+
=====*/

#ifdef ATRANSPL_H

#define ATRANSPL_H

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

#include <oratypes.h>
#include <ocidfn.h>

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

typedef struct cda_def csrdef;
typedef struct cda_def ldadef;

```

```
#ifndef NULL
#define NULL 0
#endif

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

#ifdef DISCARD
# define DISCARD (void)
#else
# define DISCARD 0
endif


#ifdef sword
# define sword int
#endif

#ifdef ub1
#define ub1 unsigned char
#endif

#define UNAME_LEN 64

#define NA          -1      /* ANSI SQL NULL */
#define VER7        2

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

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

#define OOPEN(lida,cursor)\
    if (oopen((cursor),(lida),(text*)0,NA,(text*)0,NA))\
        {sql_error(lida,cursor);} \
    else \
        DISCARD 0

#define OBNDRV(lida,cursor,sqldvar,progv,progvl,flype)\
    if (obndrv((cursor),(text\n*)(sqldvar),NA,(ub1*)(progv),(progvl),(flype),NA,\n(sdb2 *)0, (text *)0, NA, NA)))\
        {sql_error(lida,cursor);} \
    else \
        DISCARD 0

#define OPARSE(lida,cursor,sqldstm,sqli.deflg,lngflg)\
    if (oparse((cursor),(text*)(sqldstm),(sb4)(sqli).(deflg),(ub4)(lngflg)))\
        {sql_error(lida,cursor);} \
    else \
        DISCARD 0

#define OEXEC(lida,cursor)\
    if (oexec((cursor)))\
        {sql_error(lida,cursor);} \
    else \
        DISCARD 0

#define OCOM(lida,cursor)\
    if (ocom((cursor)))\
        {sql_error(lida,cursor);} \
    else \
        DISCARD 0

#define OROL(lida,cursor)\
    if (orol((cursor)))\
        {sql_error(lida,cursor);} \
    else \
        DISCARD 0

#define ODEFIN(lida,cursor,pos,buf,bufl,flype,scale,indp,fmt,fmtl,fmitt,rlen,rcode e)\
    if (odefif((cursor),(pos),(ub1*)(buf),(bufl),(flype),(scale),(indp),(text*)(fmt),(fmtl),(fmitt),(rlen),(rcode))))\
        {sql_error(lida,cursor);} \
    else \
        DISCARD 0

#define ISOTXT "alter session set isolation_level = serializable"

#define SQLTXT1 "BEGIN SELECT 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,_lpkey, _lskey, :l_quan, :l_newquan, :l_tax, :l_disc, :l_eprice, :l_neweprice, _lo_tprice, :o_newtprice, :rprice, :cost); END;"

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

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

#endif /* ATRANSPL_H */
```

```
-----+-----
rem       Copyright (c) 1995 Oracle Corp., Redwood Shores, CA
rem               OPEN SYSTEMS PERFORMANCE GROUP
|
rem           All Rights Reserved                               |
rem
=====+=====
rem FILENAME
rem   atrans.sql
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
=====
=====
```

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

```
CREATE OR REPLACE PACKAGE d_atrans
IS
PROCEDURE doatrans
(
    l_key          IN OUT number,
    o_key          IN OUT number,
    delta          IN OUT number,
    l_pkey         IN OUT number,
    l_skey         IN OUT number,
    l_quan         IN OUT number,
    l_newquan      IN OUT number,
    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 number,
    o_key          IN OUT number,
    delta          IN OUT number,
    l_pkey         IN OUT number,
    l_skey         IN OUT number,
    l_quan         IN OUT number,
    l_newquan      IN OUT number,
    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
    ottotal 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_linenum = l_key;

        ottotal := o_tprice - trunc((trunc((l_eprice * (1.0-l_disc)), 2) *
(1.0+l_tax)), 2);

        if l_quan = 0 then
            rprice := 0;
        else
            rprice := trunc((l_eprice/l_quan), 2);
        end if;

        cost := trunc((rprice * delta), 2);
        l_neweprice := l_eprice + cost;
        o_newtprice := trunc((l_neweprice * (1.0 - l_disc)), 2);
        o_newtprice := ottotal + 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_linenum = 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;
```

A_QUERY.SQL

```
rem
rem
=====
=====+
rem      Copyright (c) 1996 Oracle Corp. Redwood Shores, CA      |
rem      OPEN SYSTEMS PERFORMANCE GROUP                          |
|
rem      All Rights Reserved                                      |
rem
=====
=====+
rem  FILENAME
rem    a_query.sql
rem  DESCRIPTION
rem    Performs ACID Query for TPC-D benchmark.
rem    Asks user to input values for o_key
rem
=====
=====
```

```
rem
rem Usage: sqlplus tpcd/tpcd @a_query <o_key>
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;
```

Appendix F: Initial Ten Rows

O_ORDERDA O_ORDERKEY O_CUSTKEY O_ORDERPRIORITY O_SHIPPRIORITY O_CLERK O_O_TOTALPRICE

O_COMMENT

05-AUG-93 443450756 14435132 4-NOT SPECIFIED 0 Clerk#000026529 F 108859.95
i14Bgh24ywO2 6yCjA4yQi3S2kgiSC Shg0hhi4zjkwNNkw0N2BBjnOyOOh63

09-AUG-93 443451074 5025247 1-URGENT 0 Clerk#000010559 F 140512.98
LhS6 N7zPCLmLOgzOn003xk4ygiCnh6QQjB6O

12-AUG-93 443451168 7494622 2-HIGH 0 Clerk#000060014 F 131306.62
AyOCx6Bjgkn16SBAIjk 14BmCkLCljxQPN7PM76inzj5OPinC7jnxSy73xOiOO25

10-AUG-93 443451713 8401565 3-MEDIUM 0 Clerk#000048394 F 94715.59
41z5PAIm7IP1713g2wQ5jLxgAB3CMnxln7AySlni7kl5hB651

06-AUG-93 443451715 7442096 4-NOT SPECIFIED 0 Clerk#000082675 F 277633.16
CL0Q7njhiCMOQjgNj45zgBM7Qk210BxANwO3PhMQjkhQnjMO PB6

25-AUG-93 443451842 12361975 3-MEDIUM 0 Clerk#000079333 F 174519.54
wz0AxRmS2nKL7mx0m6My0nQLkznOyj277lmwLmnAzl1ChjCO4Lgw2 MwjPBL0MkSAAi0P

29-AUG-93 443452384 7946129 5-LOW 0 Clerk#000001183 F 259716.3

O_ORDERDA O_ORDERKEY O_CUSTKEY O_ORDERPRIORITY O_SHIPPRIORITY O_CLERK O_O_TOTALPRICE

O_COMMENT

mx8B71znP110xLnPhxQmLxhR63S4w31OmihnxnnzPkkgn0i3N7zk

19-AUG-93 443453380 14988760 3-MEDIUM 0 Clerk#000042030 F 199857.76
lgM1mOxw6z4lSIgNLLPlxL37PBR65yl0ngMOBQ1 PBh5nk2 iyylmMgQ7z644QO2MNI

15-AUG-93 443453888 9335833 2-HIGH 0 Clerk#000029283 F 152911.03
kMxgMwAhAgmn Ah3Q3BM 60LMg3L3

22-AUG-93 443453985 11493703 4-NOT SPECIFIED 0 Clerk#000040082 F 90120.79
SmLAIOR1RA 2NN4jSNywbj

10 rows selected.

L_SHIPDAT L_ORDERKEY L_DISCOUNT L_EXTENDEDPRICE L_SUPPKEY L_QUANTITY L_PARTKEY L_L_TAX

L_COMMITD L_RECEIPT L_SHIPMODE L_LINENUMBER L_SHIPINSTRUCT

L_COMMENT

08-JUN-93 24741345 .03 13227.27 867831 7 R 4367822 F .03
03-MAY-93 09-JUN-93 AIR 3 NONE
M4Nwzg7444m0Sxl

16-JUN-93 24741537 .02 25454.26 128932 13 A 17378880 F 0
29-MAR-93 05-JUL-93 FOB 3 NONE
A1g4ki3 11ynLy123Qxck5P7R3L1kn45i

26-JUN-93 24741830 .09 77310.99 861839 43 A 17111787 F .08
08-JUN-93 25-JUL-93 TRUCK 2 COLLECT COD
43yzmASBz5yi P1h Ni

08-JUN-93 24741894 0 32986.56 110076 32 A 4860071 F .04
01-AUG-93 30-JUN-93 RAIL 1 COLLECT COD
2h4MgC30AmIn03PQOiy7L P7

11-JUN-93 24742054 .03 22506.96 936151 21 R 7436136 F .07

L_SHIPDAT L_ORDERKEY L_DISCOUNT L_EXTENDEDPRICE L_SUPPKEY L_QUANTITY L_L_PARTKEY L_L_TAX

L_COMMITD L_RECEIPT L_SHIPMODE L_LINENUMBER L_SHIPINSTRUCT

L_COMMENT

11-APR-93 15-JUN-93 FOB 5 DELIVER IN PERSON
C4n5l1mNAyxB1QRBx

26-JUN-93 24742210 .05 30652.32 733351 24 A 3233344 F .04
14-JUN-93 13-JUL-93 REG AIR 2 COLLECT COD
i12yCML0mOmzknNC7j zh216Ch3B7RMxzhMmxQ0

10-JUN-93 24742210 .05 24699 554749 15 A 2804742 F .04
10-MAY-93 12-JUN-93 FOB 3 TAKE BACK RETURN
Og7CCBn0QQ

02-JUN-93 24742212 .03 33422.65 643137 31 R 19643136 F .06
23-JUL-93 03-JUN-93 FOB 3 COLLECT COD
Ah2jiCyiNMN7k4646hxnimjB2A7xLMAyzR

25-JUN-93 24742212 0 92472.5 774833 50 A 7524825 F .04
26-JUL-93 18-JUL-93 REG AIR 5 NONE

L_SHIPDAT L_ORDERKEY L_DISCOUNT L_EXTENDEDPRICE L_SUPPKEY L_QUANTITY L_L_PARTKEY L_L_TAX

L_COMMITD L_RECEIPT L_SHIPMODE L_LINENUMBER L_SHIPINSTRUCT

L_COMMENT

wl ORyO1013mwBI6OzNO2hQSI6y B

12-JUN-93 24742436 .03 38976.87 181376 29 R 6681363 F .02
26-JUN-93 09-JUL-93 TRUCK 2 NONE
nSgMAm3ML1BO3yAimR3z6nmAQng53 kzkCASglk51l

10 rows selected.

C_CUSTKEY C_MKTSEGME C_NATIONKEY C_NAME C_ADDRESS

C_PHONE C_ACCTBAL

C_COMMENT

750016 AUTOMOBILE 1 Customer#000750016 kg iPBxgk5MR4LSnyn nnL4SwzLBR.Swm3i6Rz
11-855-942-5002 9805.91
Ow24RPxAgnP77C4Pz0 gl0BkRNOI07ykBk37w4PmNBggi3P7BMhk0 zk2RCP6Q Ah7 My7SRPBQM4AzCByBILS54PANNPm3k B7

750030 AUTOMOBILE 3 Customer#000750030 1y3hSin7z0wnPLj3Sjmi0khM
13-656-631-5882 4353.7
nQnR AOP SR 2k0OzLlk4l7O202Ryy 3iM5nzglh7S2jOAN0ykN1mmw51AyLMAmkmONxhO67B76

750031 AUTOMOBILE 8 Customer#000750031 50NMh3 4jiM02 LihCMjL7z5giIMMy62h5
18-478-594-6495 9670.07
mjmLokP17MkNPBwh0AwBNC33A5P6 kLSMMRzy1w67P74P56OmMw5i3nCMIjNnAMAN6RAkihSMliAySP4Biwk3M4whPhy MgcQA
4zjQkQ36MChy3Pj

750033 AUTOMOBILE 23 Customer#000750033 Cl RIL7PQ57kz

Compaq

33-775-625-7591 4590.93
6M3x6 A5hxLjxi6Cwwm2RwNzhBi0S

C_CUSTKEY	C_MKTSE	GME	C_NATIONKEY	C_NAME	C_ADDRESS

C_PHONE	C_ACCTBAL				

C_COMMENT					

750048	AUTOMOBILE	0	Customer#000750048	7mhin52NywynBziAh	A0k1m6hmm07B
10-430-119-5177 5165.71					
2OLw 5CmjA7564j55jMgQ36ng x2hx2BCOnBB6PxB4O2C6kC25RChL76nygS 0yQAx5SyiR1CLLLMihAzQNNlwgzMB02lO3LnML1z0					
750055	AUTOMOBILE	19	Customer#000750055	5hlw530BA6mB4jixONm1mi5hBNAALOMQ	
29-743-981-6843 7630.28					
SS CzSOjABO443xQN4jgzkixBC4N2kQglzjSCIAz6nRhy zxiQ6C5Lxk4ORLQy4ROmAxA4Q0m2AgQ5k2BSwOmS5S70nCR					
750057	AUTOMOBILE	24	Customer#000750057	g2IMRwABgylBzi4ikLgRzSiLmzl	
34-862-289-9671 7732.13					
g1LnhOA1hlnhxLmQy17MSNmBzh7N17Bh67lzSn61y4AORRzKAjlC7COznRxmzgBy 7 LxiS46B0AN4R0mmCPARQjR					
750059	AUTOMOBILE	1	Customer#000750059	1S5AmkzxiPLQQi52i07gQykg	
11-418-492-2773 -988.27					
zQPmkP5Bj4miihz3RxNgOxxBN123kL xmlR0hPz2y2Bj4g1nn6LkBm5mh63yxOwP0Cn50 w60gzM41L4SR0y2ixj43QN					

C_CUSTKEY	C_MKTSE	GME	C_NATIONKEY	C_NAME	C_ADDRESS

C_PHONE	C_ACCTBAL				

C_COMMENT					

750064	AUTOMOBILE	19	Customer#000750064	0IROk nhmk4yz3S 5wl25PARj23PhNx	
29-939-733-6078 18.1					
y6N1nmSPhzk jnPllNxyw6nx6053CLkgzNm1A5Q20 B 3SM4Lm26Lz0Qx316ILQnNA0lCQC Q3MOCwxjyj m5y5LLmikkhQ6yNB33y5glIPQy67SnP					
750072	AUTOMOBILE	5	Customer#000750072	m2PzARMiB4iCgBhyQilmP7Nji	
15-665-572-9680 538.62					
m1w5z5Q2PjyhS2Qwjw7hi522w6Ay6niz 3S2S42 62522xLQ 0zL2CgmGLQx540xMwk4M1glSRx2h0mPj641SSh7ghBCnCm14z					

10 rows selected.

P_PARTKEY	P_TYPE	P_SIZE	P_BRAND

P_NAME	P_CONTAINE P_MFGR		

P_RETAILPRICE P_COMMENT			

9067482	MEDIUM POLISHED TIN	48	Brand#31
metallic mint olive yellow MED DRUM Manufacturer#3			
1449.03 MB70x 3yzSCm0A4Rz6Qyn			
9067503	PROMO BURNISHED BRASS	48	Brand#53
gainsboro coral floral sandy olive LG PKG Manufacturer#5			
1470.05 16Lm4gBBgOk6MIB62Ax			
9067576	MEDIUM ANODIZED TIN	48	Brand#24
firebrick blush snow rosy azure SM JAR Manufacturer#2			
1543.12 ABi7z			
9067608	LARGE POLISHED STEEL	48	Brand#35
lemon red antique gainsboro burnished MED PACK Manufacturer#3			
1575.15 746Om26w			

9067795 MEDIUM ANODIZED COPPER 48 Brand#55

P_PARTKEY P_TYPE P_SIZE P_BRAND

P_NAME P_CONTAINE P_MFGR

P_RETAILPRICE P_COMMENT

goldenrod khaki purple yellow ivory MED JAR Manufacturer#5
1762.34 Ill1mxkQm6RxwyCR

9067842 STANDARD BURNISHED STEEL 48 Brand#23
green turquoise khaki blached MED PACK Manufacturer#2
1809.39 Nx5 QMLnkL4MMM316

9067997 MEDIUM PLATED COPPER 48 Brand#41
maroon puff almond burlywood cyan SM PACK Manufacturer#4
1964.54 1MjI00

9068055 MEDIUM BURNISHED STEEL 48 Brand#42
red green thistle grey ivory SM CASE Manufacturer#4
1022.6 j3Ahhj4yhn

9068121 SMALL POLISHED TIN 48 Brand#24
lavender pale ghost floral JUMBO BAG Manufacturer#2

P_PARTKEY P_TYPE P_SIZE P_BRAND

P_NAME P_CONTAINE P_MFGR

P_RETAILPRICE P_COMMENT

1088.67 iyQ2hLR7iACQ

9068133 PROMO ANODIZED NICKEL 48 Brand#53
midnight violet frosted white metallic JUMBO JAR Manufacturer#5
1100.68 mhxwm05CB3

10 rows selected.

S_SUPPKEY S_NATIONKEY

S_COMMENT

S_NAME S_ADDRESS S_PHONE S_ACCTBAL

520842 24
NLwAASl0SiklxCA332zA iABi
Supplier#000520842 Qg337Lmn6yin 34-496-593-1234 1549.75

520843 5
PMB4MNxMg6B4OOSm3B1P6w3RknPiy 6hkPm3nPR
Supplier#000520843 1x30QLwx0SQS B04Cl2wyiM 15-494-209-8828 7547.94

520844 13
zNNA0SRNMSoH5z5AQlwO B2PRMmRLyMLniPB0nSmIOA 1ChmLO553nziz0yMgzj4L2j3ng45 3mRL2AgyOhzA
Supplier#000520844 wzPOjQ7PB5MCBi1AM 23-820-857-1327 4357.58

520845 10
4PyMR6APA41mQ2lwQPngwRyxRxzP7zklAjj3h4S11P37NgmgNgMnyzQzP4nSB6B0BR A3PA571iO2k23AmBlk7y7 xPy3R3A7SN1
Supplier#000520845 211izzRCxQQ5M11NLCC4NwMNMN77B 20-412-767-1918 8840.6

520846 16

Compaq

S_SUPPKEY S_NATIONKEY

S_COMMENT

S_NAME S_ADDRESS S_PHONE S_ACCTBAL

nO42QM3RjxnzxC3RyxkQABjy1yRynBjzR1xmgPN6L0Azl0O2l
Supplier#000520846 QwO7QnQCw1xCxMRj56OSLP6Sz6hhSQiNkL 26-761-104-8974 4953.73
520847 15
IB7iPL1w1NQ1SilxyQlxmR7Lyz772xwyxPBB23kzBjRnR5ljBIC5LRyR2C3MQQBzz4Pz2Qy CwkgL1L14CMy11QA2
Supplier#000520847 R NNA2w1QlPzSk761kS6khm2w3nPjmlWyPBR 25-334-536-3864 7355.48

520848 6
3A7CLi4y 6C6k45k wLg2kSAjPC2hMCnNI05imh67iSCRL73QCz4xn013PAixC 7620k2Pw257C 0LMj5jPnRnm
Supplier#000520848 BC3jL5420Nm52 P4SkALQ 16-790-281-2508 3614.59

520849 4
N6NOSnwgzSnSxOL6C6Ng2NhjzmlLP1zlLB4Qjy4CLyNB
Supplier#000520849 kzw6zAB5CCwh 14-425-432-5602 6393.1

520850 20
kzngmwLzBNkg2h 0Ank3Oz1iO04nn2gi4SAhNjRkyCiRR5Pz3wN3LmkjziPLkzn1I2

S_SUPPKEY S_NATIONKEY

S_COMMENT

S_NAME S_ADDRESS S_PHONE S_ACCTBAL

Supplier#000520850 272hRhSkB0R41SmLM01gnnww70Akjx1Bm1hP 30-787-183-1075 9892.2

520851 11
g47w B5xLxg03M02l454PhmQCAAn53whNAzmy 65iL0RQRzIAGM57BIRxBI ALxMhBiLMak
Supplier#000520851 hwkgm13h6ML65nhP1N3iAwLOP3Ch04AOBmNO 21-178-304-1158 2348.46

10 rows selected.

PS_PARTKEY PS_SUPPKEY PS_SUPPLYCOST PS_AVAILQTY

PS_COMMENT

2000001 2 962.92 1806
4MQmhCOPRICQPi1nxBxzChy32 6iSBCMRy S5Rmy0Sk4RA1QMmy7AP yzCw L5jw2l7P2Rz4h4Qkj0kC4SijmSCinMmy3k01ww5
hChg74NiRjRy2n7h z21fw42Oj115i5iCk5hB

2000001 250004 45.51 2562
P3g5PwPgX h72gmgmgnwh1 yO1RSjMNM MC4ijyLR1RILLMmMhkCAzh7yiQM00ni3

2000001 500006 883.32 9746
kx35jCzlnk5C4 i6QLBO4m4hQzghA2N2yL11jAPkkjg7Q005gixCSQ1hkIMmC M13R5 1SnnhAOgMzwmX0Nh05khgm5wBng0wAgj
mPx0mz 57Rln00SmB3Mn0Rxxw020nAS6MxOwNzglkRN5QLLglik gj

2000001 750008 11.84 4197
7Bz2hwxhR0Q NjKABROxgCOIPOw2ix33R1Ak1QgNkyB0Ahm00QMzCPw6yh6n4AwShOjmMSCOmCyj67m3wBxnnljBMMNBki kknh
yi6j4NPPB4QNmgYORm76miRwl

2000002 3 474.36 9615
zOjP5xNkyMPn07Ow3g03PPjAnNlm7iNyRhMmymS2 RgBzwnR0zlx1j5NmR 00LAn7h5ygAgM jM20i2hjiyijP 502Bm4xQ4SLCzn
IC4kx3y7Lix0gSSwhAyLLmL3Mm520zjMLBz6mh0Mgxh61R7nPCLCySQMNS y7PgZ3P

PS_PARTKEY PS_SUPPKEY PS_SUPPLYCOST PS_AVAILQTY

PS_COMMENT

2000002 250005 730.65 9802
h5nP7hQONSN Pn6nlggiLx5B3k2 3wOgihlyzm3j4nxM2yizCiS61Nm g6g1xMgmy7 5 PMy wLz M61RPRCCNPylP37SPj6SiLmQ
0PLyQz20iO6hRxOBikjzS073Q076knMRxB1jjBMLmiz7zLSCSPRAjPRMkCiNLSy5RymSxyRyy74 y4i5N16SMBh

2000002 500007 544.82 9993
5l1BQyOOjx3 BnN0POPxyOPS02mA4M1hPyy1m2hOC0Nz4g52Lm7i6wL1INyyC2572Ni5NxMi64njzRxyg3P2R0jMnyMLwwm246Pz
MP1kNC0NLSjBjRgQw0Ri7Lz0in1M2Nh7xAICyBw52L0xx

2000002 750009 75.75 852
AlngWBzLR2B7N1NSPA0n64gB134mAk3nPAjBB NL6BiyAP C1g1m0mNM1zlxPANmN0kQlgmC046yg5jw 4g2Oj jSx34lkOOnLCn
gSC4ymi0m5Q3wiR6xgC2Ry Lxk

2000003 4 660.09 6383
g1BP 06N Cmi0Lg3kxP2MS5Bnnxg0Ojkyn4w4h67Rh6ix7AhkAmLnA2injBOjIwngLROgOi6ANRLSA5mQM3NwPxPR6QmRk6n5i3
1xk0mh21x1BAk7AmS1QiQP7yR1xh62j

2000003 250006 359.27 9499
wmjyjAyLPA6CBCBQxmAnhSxwiCwxQkm3512PL5mx6M1h1wzyn3NPmz7ONMn3SCBRhBy1NiP1L3PkwS7j0QA1SjQOI3QgAmk1Qk04
QB i2 zy jwh6Ci7QC17OkngkB3

PS_PARTKEY PS_SUPPKEY PS_SUPPLYCOST PS_AVAILQTY

PS_COMMENT

10 rows selected.

N_NATIONKEY	N_NAME	N_REGIONKEY
0	ALGERIA	0
2CxhI7 L1iwk6hMh300izngN32CPwCikyLk6khMzSRA		
1	ARGENTINA	1
zQn3Okwz1wLn7PLS3OhCgn56kP5PyRikgi1B7IL		
2	BRAZIL	1
gLmS0nACAmnBCj2klki7RCPNgPxnCOjNg4k OiAg57COSOm1NwCnOyLx40R SC y20gPPAkNk5hxRhR5OmgS1iPQQzNAXPL30n67 OgyC i617Sh4LS		
3	CANADA	1
4yMO AhnQ5Lh wzQAM662Aw1ByCi7CxmzRwNR5nAIO4 x		
4	EGYPT	4
I1im5126 Cxj NMQmLxOikni02j2m3Ah4yNR1QQiL507j2QSlyN		
5	ETHIOPIA	0
NS7n LSOP Oz5n1AIB2S02hN0IMh4SBxP iRhBO 047R26 2BIM		

N_NATIONKEY	N_NAME	N_REGIONKEY
6	FRANCE	3
3mjimizl S 3L3k2hNNhNIP4w370xRxyN15wn		
7	GERMANY	3
z nOP4RkwO CmzBB 516mAg iByw4OM3QyNPA		
8	INDIA	2
MNIR5RCiRMjI111wjN7Myn M1IyINIMmBQI7PL4C kKxQkgPQ7i3w6B67R2QkOO40xI4Q2iw76jRL7ihR5Q 0xC7RRm5iQ2NAX 2LiBm3QiO27j		

9 INDONESIA 2
SjPmQQ71Lj 7ABj6MxIAQk3nLwi73BPxzCwjzMn4zILzgg6nnz0j0w zxC66gP6ykRPMg

10 rows selected.

R_REGIONKEY R_NAME

R_COMMENT

0 AFRICA
xSx31zz31Ci1z4OAnmm05AjiOxC3AMMNOgC0kACgwngg3glP7LLLywlQy7R

1 AMERICA
kgyh3LSnC72k6zIAz0LP3k2L4QB1QL1O673OjO1SPj0ngQ7CO100SBgmgRQ4lgPCmk21A425iklyAR4yBRAwR4Cm5miNw 4jl13m
Mnxw17B

2 ASIA
NSg6xlMIA1lzm6mOR0Ajsx nhRA77NgRxBwL1M6Py RjySB3RLwkyPkwMM2R1BQ xAzkOgkjmI0gAghinP5inmNmR76MlijMS3S2
zxONR15

3 EUROPE
zlSL7Qwg12hMBL5ihlz0M45QkjShwSyiO04MLOh7wn1ARLQPyPayAiil5761Li7AInR1S RQ4SLny7B2Ryj5P66MLhn Nxhwb4C
3ig0SO

4 MIDDLE EAST
RIlxmhPLz3Cy2mNlg4QMBnNASMACKi MPki7Oi

Appendix G: DBGEN Source Code

Changes to DBGEN v.1.2.0 are in the following files.

```
DRIVER.C
/* Scsid:  @(#)driver.c          9.1.1.34   5/1/96  11:45:07 */
/* main driver for dss benchmark */

#define DECLARER      /* EXTERN references get defined here */
#define NO_FUNC(int (*)) NULL /* to clean up tdefs */
#define NO_LFUNC(long (*)) NULL /* to clean up ldefs */
#define LIFENOISE(cnt) if (i % cnt == 0 && verbose) fprintf(stderr, ".")

#include "config.h"
#include <stdlib.h>
#if (defined(_POSIX_)) || defined(WIN32) /* Change for Windows NT
*/
#include <unistd.h>
#include <sys/wait.h>
#endif /* WIN32 */
#include <stdio.h> /* */
#include <limits.h>
#include <math.h>
#include <ctype.h>
#include <signal.h>
#include <string.h>
#include <errno.h>
#ifdef HP
#include <strings.h>
#endif
#if (defined(WIN32) && defined(_POSIX_))
#include <process.h>
#include <signal.h>
#pragma warning(disable:4201)
#pragma warning(disable:4214)
#pragma warning(disable:4514)
#define WIN32_LEAN_AND_MEAN
#define NOATOM
#define NOGDICAPMASKS
#define NOMETAFILE
#define NOMINMAX
#define NOMSG
#define NOOPENFILE
#define NORASTEROPS
#define NOSCROLL
#define NOSOUND
#define NOSYMETRICS
#define NOTEXTMETRIC
#define NOWH
#define NOCOMM
#define NOKANJI
#define NOMCX
#include <windows.h>
#pragma warning(default:4201)
#pragma warning(default:4214)
#endif

#include "dss.h"
#include "dsstypes.h"
#include "bcd2.h"

/*
 * Function prototypes
 */

void usage(void);
int prep_direct(char *);
int close_direct(void);
void kill_load(void);
int pload(int tbl);
void gen_tbl(int tnum, long start, long count, long upd_num);
int pr_drange(int tbl, long min, long cnt, long num);
int set_files(int t, int pload);
void seed_name(char *tgt, long s, long c, long p);
int partial(int, int);

#ifdef ORIG
#include "part.h"
int partition = 0; /* partition flag */
vals *values;
FILE *in; /* fdes for input file */
int *out; /* output fd array */
FILE *par; /* fdes for params file */
char param[FLEN];
extern char outname[];
extern FILE *outparam; /* fd for output parameter file */
int use_outparam = 0; /* flag to designate whether to use outparam
file */
long numpart = 0; /* number of partitions */
long partnum; /* partition number */
int numcol = 0; /* number of columns */
char delim = '|'; /* flatfile delimiter */
#endif /* !ORIG */

extern int optind,
opterr;
extern char *optarg;
long rowcnt = 0,
minrow = 0,
upd_num = 0;
double flt_scale;
#if (defined(WIN32) && !defined(_POSIX_))
char *spawn_args[25];
#endif

/*
 * general table descriptions. See dss.h for details on structure
 * NOTE: tables with no scaling info are scaled according to
 * another table
 *
 * the following is based on the tdef structure defined in dss.h as:
 * typedef struct
 * {
 * char *name; /* name of the table;
 * flat file output in <name>.tbl
 * long base; /* base scale rowcount of table;
 * 0 if derived
 * int (*header) (); /* function to prep output
 * int (*loader[2]) (); /* functions to present output
 * long (*gen_seed) (); /* functions to seed the RNG
 * int child; /* non-zero if there is an associated
 * detail table
 * } tdef;
 */
```

```

*/

/*
 * flat file print functions; used with -F(lat) option
 */
int pr_cust(customer_t * c, int mode);
int pr_line(order_t * o, int mode);
int pr_order(order_t * o, int mode);
int pr_part(part_t * p, int mode);
int pr_psupp(part_t * p, int mode);
int pr_supp(supplier_t * s, int mode);
int pr_order_line(order_t * o, int mode);
int pr_part_psupp(part_t * p, int mode);
int pr_time(dss_time_t * t, int mode);
int pr_nation(code_t * c, int mode);
int pr_region(code_t * c, int mode);

/*
 * inline load functions; used with -D(irect) option
 */
int ld_cust(customer_t * c, int mode);
int ld_line(order_t * o, int mode);
int ld_order(order_t * o, int mode);
int ld_part(part_t * p, int mode);
int ld_psupp(part_t * p, int mode);
int ld_supp(supplier_t * s, int mode);
int ld_order_line(order_t * o, int mode);
int ld_part_psupp(part_t * p, int mode);
int ld_time(dss_time_t * t, int mode);
int ld_nation(code_t * c, int mode);
int ld_region(code_t * c, int mode);

/*
 * seed generation functions; used with '-O s' option
 */
long sd_cust(long skip_count);
long sd_line(long skip_count);
long sd_order(long skip_count);
long sd_part(long skip_count);
long sd_psupp(long skip_count);
long sd_supp(long skip_count);
long sd_order_line(long skip_count);
long sd_part_psupp(long skip_count);
long sd_nation(long skip_count);
long sd_region(long skip_count);

/*
 * header output functions; used with -h(eader) option
 */
int hd_cust(FILE * f);
int hd_line(FILE * f);
int hd_order(FILE * f);
int hd_part(FILE * f);
int hd_psupp(FILE * f);
int hd_supp(FILE * f);
int hd_order_line(FILE * f);
int hd_part_psupp(FILE * f);
int hd_time(FILE * f);
int hd_nation(FILE * f);
int hd_region(FILE * f);

typedef      tdefs[] =
{
    {"part.tbl", "part table", 200000, hd_part,
     {pr_part, ld_part}, sd_part, NONE},
    {"partsupp.tbl", "partsupplier table", 200000, hd_psupp,
     {pr_psupp, ld_psupp}, sd_psupp, NONE},
    {"supplier.tbl", "suppliers table", 10000, hd_supp,
     {pr_supp, ld_supp}, sd_supp, NONE},
    {"customer.tbl", "customers table", 150000, hd_cust,
     {pr_cust, ld_cust}, sd_cust, NONE},
    {"order.tbl", "order table", 150000, hd_order,
     {pr_order, ld_order}, sd_order, NONE},
    {"lineitem.tbl", "lineitem table", 150000, hd_line,
     {pr_line, ld_line}, sd_line, NONE},
    {"order.tbl", "order/lineitem tables", 150000, hd_order_line,
     {pr_order_line, ld_order_line}, sd_order_line, LINE},
    {"part.tbl", "part/partsupplier tables", 200000, hd_part_psupp,
     {pr_part_psupp, ld_part_psupp}, sd_part_psupp, PSUPP},
    {"time.tbl", "time table", 2557, hd_time,
     {pr_time, ld_time}, NO_LFUNC, NONE},
    {"nation.tbl", "nation table", NATIONS_MAX, hd_nation,
     {pr_nation, ld_nation}, sd_nation, NONE},
    {"region.tbl", "region table", NATIONS_MAX, hd_region,
     {pr_region, ld_region}, sd_region, NONE},
    {"TPCDSEED", NULL, 0, NO_FUNC,
     {NO_FUNC, NO_FUNC}, NO_LFUNC, NONE}
};

int pids[MAX_CHILDREN];

void
mk_sparse(long *low_res, long *high_res, long base, long seq)
{
    long low_mask, seq_mask, overflow = 0;
    int count = 0;

    low_mask = (1<<SPARSE_KEEP) - 1;
    seq_mask = (1<<SPARSE_BITS) - 1;
    bcd2_bcd2(base, low_res, high_res);
    bcd2_div(low_res, high_res, 1<<SPARSE_KEEP);
    bcd2_mul(low_res, high_res, 1<<SPARSE_BITS);
    bcd2_add(low_res, high_res, seq);
    bcd2_mul(low_res, high_res, 1<<SPARSE_KEEP);
    bcd2_add(low_res, high_res, base & low_mask);
    bcd2_bin(&low_mask, *low_res);
    bcd2_bin(&seq_mask, *high_res);
    *low_res = low_mask;
    *high_res = seq_mask;
    return;
}

/*
 * routines to handle the graceful cleanup of multi-process loads
 */

void
stop_proc(int signum)
{
    exit(0);
}

void
kill_load(void)
{
    int i;

    #if !defined(U2200) && !defined(DOS)
        for (i=0; i< MAX_CHILDREN; i++)
            if (pids[i])
                KILL(pids[i]);
    #endif /* U2200 && !DOS */
    return;
}

/*

```

```

* re-set default output file names
*/
int
set_files(int i, int pload)
{
    char line[80];
    *new_name;

    if (table & (1 << i))
child_table:
    {
        if (pload != -1)
            sprintf(line, "%s%d", tdefs[i].name, pload + 1);
        else
        {
            printf("Enter new destination for %s data: ",
                tdefs[i].name);
            if (fgets(line, sizeof(line), stdin) == NULL)
                return(-1);
            if ((new_name = strchr(line, '\n')) != NULL)
                *new_name = '\0';
            if (strlen(line) == 0)
                return(0);
        }
        new_name = (char *)malloc(strlen(line) + 1);
        MALLOC_CHECK(new_name);
        strcpy(new_name, line);
        tdefs[i].name = new_name;
        if (tdefs[i].child != NONE)
        {
            i = tdefs[i].child;
            tdefs[i].child = NONE;
            goto child_table;
        }
    }

    return(0);
}

/*
* read the distributions needed in the benchamrk
*/
void
load_dists(void)
{
    read_dist(env_config(DIST_TAG, DIST_DFLT), "p_cntr",
        &p_cntr_set);
    read_dist(env_config(DIST_TAG, DIST_DFLT), "colors", &colors);
    read_dist(env_config(DIST_TAG, DIST_DFLT), "p_types",
        &p_types_set);
    read_dist(env_config(DIST_TAG, DIST_DFLT), "nations", &nations);
    read_dist(env_config(DIST_TAG, DIST_DFLT), "regions",
        &regions);
    read_dist(env_config(DIST_TAG, DIST_DFLT), "o_oprio",
        &o_priority_set);
    read_dist(env_config(DIST_TAG, DIST_DFLT), "instruct",
        &l_instruct_set);
    read_dist(env_config(DIST_TAG, DIST_DFLT), "smode",
        &l_smode_set);
    read_dist(env_config(DIST_TAG, DIST_DFLT), "category",
        &l_category_set);
    read_dist(env_config(DIST_TAG, DIST_DFLT), "rflag",
        &l_rflag_set);
    read_dist(env_config(DIST_TAG, DIST_DFLT), "msegmnt",
        &c_mseg_set);
}

}

/*
* generate a particular table
*/
void
gen_tbl(int tnum, long start, long count, long upd_num)
{
    long i;
    order_t o;
    supplier_t supp;
    customer_t cust;
    part_t part;
    dss_time_t t;
    code_t code;
    long sk_low, sk_high;
    static long max_easy = LONG_MAX >> SPARSE_BITS;
    static int completed = 0;

    for (i = start; count; count--, i++)
    {
        LIFENOISE(1000);
        switch(tnum)
        {
            case LINE:
            case ORDER:
            case ORDER_LINE:
                if (i > max_easy)
                    mk_sparse(&sk_low, &sk_high, i,
                        (upd_num == 0)?0:1 + upd_num/(10000 / refresh));
                else
                {
                    sk_low = MK_SPARSE(i,
                        (upd_num == 0)?0:1 + upd_num/(10000 / refresh));
                    sk_high = 0;
                }
                mk_order(sk_low, sk_high, &o);
                tdefs[tnum].loader[direct] (&o, upd_num);
                break;
            case SUPP:
                mk_supp(i, &supp);
                tdefs[tnum].loader[direct] (&supp, upd_num);
                break;
            case CUST:
                mk_cust(i, &cust);
                tdefs[tnum].loader[direct] (&cust, upd_num);
                break;
            case PSUPP:
            case PART:
            case PART_PSUPP:
                mk_part(i, &part);
                tdefs[tnum].loader[direct] (&part, upd_num);
                break;
            case TIME:
                mk_time(i, &t);
                tdefs[tnum].loader[direct] (&t, 0);
                break;
            case NATION:
                mk_nation(i, &code);
                tdefs[tnum].loader[direct] (&code, 0);
                break;
            case REGION:
                mk_region(i, &code);
                tdefs[tnum].loader[direct] (&code, 0);
                break;
        }
    }
}

```

```

    }
    completed |= 1 << tnum;
}

void
usage(void)
{
    fprintf(stderr, "%s\n%s\n\t%s\n\t%s %s\n\n",
        "USAGE:",
        "dbgen [-v{vFD}] [-O {fmtst}] [-T {pcsoPSOL}]",
        "[-s <scale>] [-C <procs>] [-S <step>]",
        "dbgen [-v] [-O {dhmrt}] [-s <scale>]",
        "[-U <updates>] [-r <percent>]");
    fprintf(stderr, "-C <n> -- use <n> processes to generate data\n");
    fprintf(stderr, "    [Under DOS, must be used with -S]\n");
    fprintf(stderr, "-D    -- do database load in line\n");
    fprintf(stderr, "-f    -- force. Overwrite existing files\n");
    fprintf(stderr, "-F    -- generate flat files output\n");
    fprintf(stderr, "-h    -- display this message\n");
    fprintf(stderr, "-n <s> -- inline load into database <s>\n");
    fprintf(stderr, "-O d  -- generate SQL syntax for deletes\n");
    fprintf(stderr, "-O f  -- over-ride default output file names\n");
    fprintf(stderr, "-O h  -- output files with headers\n");
    fprintf(stderr, "-O m  -- produce columnar output\n");
    fprintf(stderr, "-O r  -- generate key ranges for deletes\n");
    fprintf(stderr, "-O s  -- generate seed sets ONLY\n");
    fprintf(stderr, "-O t  -- use TIME table and julian dates\n");
    fprintf(stderr, "-r <n> -- updates refresh (n/100)% of the\n");
    fprintf(stderr, "    data set\n");
    fprintf(stderr, "-R <n> -- resume seed rfile generation with step\n");
    fprintf(stderr, "-s <n> -- set Scale Factor (SF) to <n>\n");
    fprintf(stderr, "-S <n> -- build the <n>th step of the data set\n");
    fprintf(stderr, "-T c  -- generate customers ONLY\n");
    fprintf(stderr, "-T l  -- generate nation/region ONLY\n");
    fprintf(stderr, "-T L  -- generate lineitem ONLY\n");
    fprintf(stderr, "-T n  -- generate nation ONLY\n");
    fprintf(stderr, "-T o  -- generate orders/lineitem ONLY\n");
    fprintf(stderr, "-T O  -- generate orders ONLY\n");
    fprintf(stderr, "-T p  -- generate parts/partsupp ONLY\n");
    fprintf(stderr, "-T P  -- generate parts ONLY\n");
    fprintf(stderr, "-T r  -- generate region ONLY\n");
    fprintf(stderr, "-T s  -- generate suppliers ONLY\n");
    fprintf(stderr, "-T S  -- generate partsupp ONLY\n");
    fprintf(stderr, "-U <s> -- generate <s> update sets\n");
    fprintf(stderr, "-v    -- enable VERBOSE mode\n");
#ifdef ORIG
    fprintf(stderr, "\n===== \n");
    fprintf(stderr, "Additions for partitioning:\n");
    fprintf(stderr, "-p      -- enables partitioning\n");
    fprintf(stderr, "-i <param file> -- input parameter file\n");
    fprintf(stderr, "-a <out list>  -- output prefix list\n");
    fprintf(stderr, "-o <out prefix> -- output file prefix, -a option overrides\n");
    fprintf(stderr, "-o option\n");
    fprintf(stderr, "Format of parameter file\n");
    fprintf(stderr, "<delimiter>\n");
    fprintf(stderr, "<number of partitions>\n");
    fprintf(stderr, "<col1 type>\n");
    fprintf(stderr, "<col1 pos> - not used at this point\n");
    fprintf(stderr, "<col1 val1>\n");
    fprintf(stderr, "<col1 val2>\n");
    fprintf(stderr, "... \n");
    fprintf(stderr, "<col1 valm>\n");
    fprintf(stderr, "Notes:\n");
    fprintf(stderr, "%s\n",

```

```

        "Column types are: 0 for numbers, 1 for strings, 2 for
        dates");
    fprintf(stderr, "The direct load option is automatically turned on.\n");
    fprintf(stderr, "===== \n");
#endif /* ORIG */
    fprintf(stderr,
        "\nTo generate the SF=1 (1GB) database population , use:\n");
    fprintf(stderr, "\tdbgen -vF -s 1\n");
    fprintf(stderr, "\n%s %s\n",
        "To generate the qualification database population",
        "(100 MB), use:\n");
    fprintf(stderr, "\tdbgen -vF -s 0.1\n");
    fprintf(stderr, "\nTo generate updates for a SF=1 (1GB), use:\n");
    fprintf(stderr, "\tdbgen -v -O s -s 1\n");
    fprintf(stderr, "\tdbgen -v -U 1 -s 1\n");
}

/*
 * pload() -- handle the parallel loading of tables
 */
#ifdef DOS
int
partial(int tbl, int s)
{
    long rowcnt,
        minrow,
        char fname[80];

    rowcnt = tdefs[tbl].base * scale;
    if (rowcnt % children)
    {
        fprintf(stderr, "'-C' cannot split load equally\n");
        exit(-1);
    }
    else
        rowcnt /= children;
    if (verbose)
    {
        fprintf(stderr, "Starting to load stage %d of %d of %s...",
            s + 1, children, tdefs[tbl].comment);
    }
    if (load_state(scale, children, s))
    {
        seed_name(fname, scale, children, s);
        fprintf(stderr, "Unable to load seeds (%s)\n", fname);
        exit(-1);
    }
    minrow = rowcnt * s + 1;
    if (direct == 0)
        set_files(tbl, s);
    gen_tbl(tbl, minrow, rowcnt, upd_num);
    if (verbose)
        fprintf(stderr, "done.\n");

    return(0);
}

int
pload(int tbl)
{
    int c = 0,
        i,
        status;
    char cmdline[256];

    rowcnt = tdefs[tbl].base * scale;

```

```

if (rowcnt % children)
{
    fprintf(stderr, "'-C' cannot split load equally\n");
    exit(-1);
}
else
    rowcnt /= children;
if (verbose)
{
    fprintf(stderr, "Starting %d children to load %s",
        children, tdefs[tbl].comment);
}
for (c=0; c < children; c++)
{
    pids[c] = SPAWN();
    if (pids[c] == -1)
    {
        perror("Child loader not created");
        kill_load();
        exit(-1);
    }
    else if (pids[c] == 0) /* CHILD */
    {
        SET_HANDLER(stop_proc);
        verbose = 0;
        partial(tbl, c);
        exit(0);
    }
    else if (verbose) /* PARENT */
        fprintf(stderr, ".");
}

if (verbose)
    fprintf(stderr, "waiting...");

c = children;
while (c)
{
    i = WAIT(&status, pids[c - 1]);
    if (i == -1 && children)
    {
        if (errno==ECHILD)
            fprintf(stderr, "Could not wait on pid %d\n", pids[c-1]);
        else if (errno==EINTR)
            fprintf(stderr, "Process %d stopped abnormally\n", pids[c-1]);
        else if (errno==EINVAL)
            fprintf(stderr, "Program bug\n");
    }
    if (status & 0xFF)
    {
        if (status & 0xFF == 0117)
            printf("Process %d: STOPPED\n", i);
        else
            printf("Process %d: rcvd signal %d\n",
                i, status&0x7F);
    }
    c--;
}

if (direct == 0)
{
    sprintf(cmdline, "mv %s/pl%d.%d%c %s/%s",
        env_config(PATH_TAG, PATH_DFLT), pids[0], tbl,
        'B' + tdefs[tbl].child,
        env_config(PATH_TAG, PATH_DFLT), tdefs[tbl].name);
    system(cmdline);
}
#ifdef WIN32 && !defined(_POSIX_)
    sprintf(cmdline, "COPY %s\\pl*.%d %s\\%s",
        env_config(PATH_TAG, PATH_DFLT), tbl,
        env_config(PATH_TAG, PATH_DFLT), tdefs[tbl].name);

    fprintf(stderr, "%s\n", cmdline);
    system(cmdline);

    sprintf(cmdline, "DEL %s\\pl*.%d",
        env_config(PATH_TAG, PATH_DFLT), tbl);
    fprintf(stderr, "%s\n", cmdline);
    system(cmdline);
#else
    for (c=1; c < children; c++)
    {
        sprintf(cmdline,
            "cat %s/pl%d.%d%c >> %s/%s; rm %s/pl%d.%d%c",
            env_config(PATH_TAG, PATH_DFLT), pids[c], tbl,
            'B' + tdefs[tbl].child,
            env_config(PATH_TAG, PATH_DFLT), tdefs[tbl].name,
            env_config(PATH_TAG, PATH_DFLT), pids[c], tbl,
            'B' + tdefs[tbl].child);
        system(cmdline);
    }
#endif /* WIN32 */
}

fprintf(stderr, "done\n");
return(0);
}
#endif /* !DOS */

void
process_options(int count, char** vector)
{
    int option;

#ifdef ORIG
    /* set some default */
    if (partition) {
        (void) sprintf(param, "%s", "/part.param");
        (void) sprintf(outname, "%s", "/dev/sqlldr");
    }
#endif /* !ORIG */

#ifdef DOS
#ifdef ORIG
    while ((option = getopt(count, vector,
        "C:DFhn:O:P:r:R:s:S:T:U:vi:o:a:p")) != -1)
    {
        if (option == 'C')
            continue;
        if (option == 'D')
            continue;
        if (option == 'F')
            continue;
        if (option == 'h')
            continue;
        if (option == 'n')
            continue;
        if (option == 'O')
            continue;
        if (option == 'P')
            continue;
        if (option == 'r')
            continue;
        if (option == 'R')
            continue;
        if (option == 's')
            continue;
        if (option == 'S')
            continue;
        if (option == 'T')
            continue;
        if (option == 'U')
            continue;
        if (option == 'v')
            continue;
        if (option == 'i')
            continue;
        if (option == 'o')
            continue;
        if (option == 'a')
            continue;
        if (option == 'p')
            continue;
    }
#endif
#endif
}

```

```

        case 'o':
            strcpy(outname, optarg);
            break;
        case 'a':
            if ((outparam = fopen(optarg, "r")) == NULL) {
                fprintf(stderr, "Error opening file %s\n",
                    optarg);
            }
            use_outparam++;
            break;
    #endif /* !ORIG */
    case 'S': /* generate a particular STEP */
        step = atoi(optarg) - 1;
        break;
    case 'v': /* life noises enabled */
        verbose = 1;
        break;
    case 'f': /* blind overwrites; Force */
        force = 1;
        break;
    case 'T': /* generate a specific table */
        switch (*optarg)
        {
            case 'c': /* generate customer ONLY */
                table = 1 << CUST;
                break;
            case 'L': /* generate lineitems ONLY */
                table = 1 << LINE;
                break;
            case 'I': /* generate code table ONLY */
                table = 1 << NATION;
                table |= 1 << REGION;
                break;
            case 'n': /* generate nation table ONLY */
                table = 1 << NATION;
                break;
            case 'O': /* generate orders ONLY */
                table = 1 << ORDER;
                break;
            case 'o': /* generate orders/lineitems ONLY */
                table = 1 << ORDER_LINE;
                break;
            case 'P': /* generate part ONLY */
                table = 1 << PART;
                break;
            case 'p': /* generate part/partsupp ONLY */
                table = 1 << PART_PSUPP;
                break;
            case 'r': /* generate region table ONLY */
                table = 1 << REGION;
                break;
            case 'S': /* generate partsupp ONLY */
                table = 1 << PSUPP;
                break;
            case 's': /* generate suppliers ONLY */
                table = 1 << SUPP;
                break;
            case 't': /* generate time ONLY */
                table = 1 << TIME;
                break;
            default:
                fprintf(stderr, "Unknown table name %s\n",
                    optarg);
                usage();
                exit(1);
        }
        break;
    case 's': /* scale by Percentage of base rowcount */

```

```

        case 'P': /* for backward compatibility */
            flt_scale = atof(optarg);
            if (flt_scale < MIN_SCALE)
            {
                int i;

                scale = 1;
                for (i=PART; i < TIME; i++)
                {
                    tdefs[i].base *= flt_scale;
                    if (tdefs[i].base < 1)
                        tdefs[i].base = 1;
                }
            }
            else
                scale = (long)flt_scale;
    #ifndef NO_SUPPORT
        if (scale > MAX_SCALE)
        {
            fprintf(stderr, "%s %5.0f %s\n\t%s\n\n",
                "NOTE: Data generation for scale factors >",
                MAX_SCALE,
                "GB is",
                "still in development,",
                "and is",
                "not yet supported.\n");
            exit(1);
        }
    #endif /* NO_SUPPORT */
        break;
    case 'O': /* optional actions */
        switch (tolower(*optarg))
        {
            case 'd': /* generate SQL for deletes */
                gen_sql = 1;
                break;
            case 'f': /* over-ride default file names */
                fnames = 1;
                break;
            case 'h': /* generate headers */
                header = 1;
                break;
            case 'm': /* generate columnar output */
                columnar = 1;
                break;
            case 'r': /* generate key ranges for delete */
                gen_rng = 1;
                break;
            case 's': /* generate seed sets */
                bld_seeds = 1;
                break;
            case 't': /* use TIME table and join fields */
                oldtime = 1;
                if (updates == 0)
                    table |= (1 << TIME);
                break;
            default:
                fprintf(stderr, "Unknown option name %s\n",
                    optarg);
                usage();
                exit(1);
        }
        break;
    case 'D': /* direct load of generated data */
        direct = 1;
        break;
    case 'F': /* generate flat files for later loading */

```

```

        direct = 0;
        break;
    case 'U': /* generate flat files for update stream */
        updates = atoi(optarg);
        break;
    case 'r': /* set the refresh (update) percentage */
        refresh = atoi(optarg);
        break;
    case 'R': /* resume seed file generation */
        resume = atoi(optarg);
        break;
#ifdef DOS
    case 'C':
        children = atoi(optarg);
        if (children > MAX_CHILDREN)
        {
            fprintf(stderr, "Children reset to limit of %d\n",
                MAX_CHILDREN);
            children = MAX_CHILDREN;
        }
        break;
#endif /* !DOS */
    case 'n': /* set name of database for direct load */
        db_name = (char *)malloc(strlen(optarg) + 1);
        MALLOC_CHECK(db_name);
        strcpy(db_name, optarg);
        break;
    default:
        printf("ERROR: option '%c' unknown.\n",
            *(vector[optind] + 1));
    case 'h': /* something unexpected */
        fprintf(stderr,
            "TPC-D Population Generator (Version %d.%d.%d%c)\n",
            VERSION, RELEASE,
            MODIFICATION, PATCH);
        fprintf(stderr, "Copyright %s %s\n", TPC, C_DATES);
        usage();
        exit(1);
    }
    return;
}

/*
 * MAIN
 *
 * assumes the existence of getopt() to clean up the command
 * line handling
 */
int
main(int ac, char **av)
{
    int i;
    c;

    table = (1 << CUST) |
        (1 << SUPP) |
        (1 << NATION) |
        (1 << REGION) |
        (1 << PART_PSUPP) |
        (1 << ORDER_LINE);

    force = 0;
    verbose = 0;
    columnar = 0;
    bld_seeds = 0;
    header = 0;
    oldtime = 0;
    direct = 0;

    scale = 1;
    flt_scale = 1.0;
    updates = 0;
    refresh = UPD_PCT;
    resume = -1;
    step = -1;
    tdefs[ORDER].base *=
        ORDERS_PER_CUST; /* have to do this after init */
    tdefs[LINE].base *=
        ORDERS_PER_CUST; /* have to do this after init */
    tdefs[ORDER_LINE].base *=
        ORDERS_PER_CUST; /* have to do this after init */
    fnames = 0;
    db_name = NULL;
    gen_sql = 0;
    gen_rng = 0;
    children = 1;

    process_options(ac, av);

#ifdef ORIG
    if (partition) {
        if ((par = fopen(param, "r")) == NULL) {
            fprintf(stderr, "Unable to open file '%s'\n", param);
            fprintf(stderr, "%s: %s\n", param, strerror(errno));
            exit(-1);
        }
        (void) process_param_file();
    }
#endif /* ORIG */

    if (defined(WIN32)&&!defined(_POSIX_))
        for (i=0; i < ac; i++)
        {
            spawn_args[i] = malloc((strlen(av[i]) + 1) * sizeof(char));
            MALLOC_CHECK(spawn_args[i]);
            strcpy(spawn_args[i], av[i]);
        }
    spawn_args[ac] = NULL;
#endif

    fprintf(stderr,
        "TPC-D Population Generator (Version %d.%d.%d%c)\n",
        VERSION, RELEASE, MODIFICATION, PATCH);
    fprintf(stderr, "Copyright %s %s\n", TPC, C_DATES);

    load_dists();
    /* have to do this after init */
    tdefs[NATION].base = nations.count;
    tdefs[REGION].base = regions.count;

    /*
     * updates are never parallelized
     */
    if (updates)
    {
        for (i=LINE; i <= ORDER_LINE; i++)
        {
            if (table & (1 << i))
            {
                if (load_state(scale, children, children))
                {
                    fprintf(stderr, "Unable to load seeds (%d)\n",
                        scale);
                    exit(-1);
                }
            }
            rowcnt = tdefs[i].base / 10000 * scale * refresh;

```

```

    upd_num = 0;
    while (upd_num < updates)
    {
        minrow = rowcnt * upd_num + 1;
        if (verbose)
            fprintf(stderr,
                "Generating updates for %s [pid: %d]",
                tdefs[i].comment, DSS_PROC);
        gen_tbl(i, minrow, rowcnt, upd_num + 1);
        if (verbose)
            fprintf(stderr, "done.\n");
        pr_drangle(i, minrow, rowcnt, upd_num + 1);
        upd_num++;
    }
}
exit(0);
}
/*
 * seed gen is never parallelized
 */
if (bld_seeds)
{
    if (resume == -1)
    {
        c = 1;
    }
    else
    {
        c = resume;
        if (load_state(scale, children, c-1))
        {
            char fname[80];
            seed_name(fname,
                scale, children, c-1);
            fprintf(stderr, "Unable
                to load seeds (%s)\n", fname);
            exit(-1);
        }
    }
    for (; c <= children; c++)
    {
        for (i = PART; i <= PART_PSUPP; i++)
            if (table & (1 << i))
            {
                if ((tdefs[i].base * scale % children) == 0)
                    rowcnt = tdefs[i].base * scale / children;
                else
                {
                    fprintf(stderr, "Load cannot be evenly split!");
                    exit(1);
                }
                if (verbose)
                    fprintf(stderr,
                        "Generating seeds for %s [step: %d] ",
                        tdefs[i].comment, c);
                tdefs[i].gen_seed(rowcnt);
                if (verbose)
                    fprintf(stderr, "done.\n");
            }
        if (store_state(scale, children, c))
        {
            fprintf(stderr, "Unable to store seeds\n");
            exit(-1);
        }
    }
}
exit(0);
}

/**
 ** actual data generation section starts here
 **/
/*
 * open database connection or set all the file names, as appropriate
 */
if (direct)
    prep_direct((db_name)?db_name:DBNAME);
else
    if (frames)
        for (i = PART; i <= REGION; i++)
        {
            if (table & (1 << i))
                if (set_files(i, -1))
                {
                    fprintf(stderr, "Load aborted\n");
                    exit(1);
                }
        }

    for (i = PART; i <= REGION; i++)
        if (table & (1 << i))
        {
            if (children > 1 && i < TIME)
                if (step >= 0)
                    partial(i, step);
        }
#ifdef DOS
    else
    {
        fprintf(stderr,
            "Parallel load is not supported on your platform.\n");
        exit(1);
    }
#else
    else
        pload(i);
#endif /* DOS */
    else
    {
        minrow = 1;
        if (i < TIME)
            rowcnt = tdefs[i].base * scale;
        else
            rowcnt = tdefs[i].base;
        if (verbose)
            fprintf(stderr, "Generating data for %s [pid: %d]",
                tdefs[i].comment, DSS_PROC);
        gen_tbl(i, minrow, rowcnt, upd_num);
        if (verbose)
            fprintf(stderr, "done.\n");
    }
}

if (direct)
    close_direct();

return (0);
}

LOAD_STUB.C
/*****
 * Title:    load_stub.c
 * Scsid:    @(#)load_stub.c      9.1.1.11   9/7/95 16:24:05
 * Description:

```

```

*      stub routines for:
*      inline load of dss benchmark
*      header creation for dss benchmark
*
*****
*/

#include <stdio.h>
#include "config.h"
#include "dss.h"
#include "dsstypes.h"
#include "part.h"

#include <io.h>

extern vals *values;
extern int *out;      /* output fd array      */
extern long partnum;  /* partition number      */
extern int partition;
char tmpdate[16];     /* temporary storage to store dates */
extern void Exitprog();

int
close_direct(void)
{
    /* any post load cleanup goes here */
    return(0);
}

int
prep_direct(void)
{
    /* any preload prep goes here */
    return(0);
}

int
hd_cust (FILE *f)
{
    static int count = 0;

    if (! count++)
        printf("No header has been defined for the customer table\n");

    return(0);
}

int
ld_cust (customer_t *c, int mode)
{
    char line[256];
    char *ptr;

    if (partition) {
        partnum = find_partition(c->mktsegment, 1);
    }
    PPR_START(line, ptr);
    PPR_INT(ptr, c->custkey);
    PPR_STR(ptr, c->mktsegment, C_MSEG_LEN);
    PPR_INT(ptr, c->nation_code);
    PPR_VSTR(ptr, c->name, C_NAME_LEN);
    PPR_VSTR(ptr, c->address,
              (columnar)?(long)(ceil(C_ADDR_LEN *
V_STR_HGH)):c->alen);
    PPR_STR(ptr, c->phone, PHONE_LEN);
    PPR_MONEY(ptr, c->acctbal);
    PPR_VSTR(ptr, c->comment,
              (columnar)?(long)(ceil(C_CMNT_LEN * V_STR_HGH)):c-
>clen);
    PPR_END(ptr);

    if (write(out[partnum], line, strlen(line)) == -1) {
        fprintf(stderr, "Error when writing customer data\n");
        Exitprog(errno);
    }

    return(0);
}

int
ld_part (FILE *f)
{
    static int count = 0;

    if (! count++)
        printf("No header has been defined for the part table\n");

    return(0);
}

int
ld_part (part_t *pp, int mode)
{
    char line[256];
    char *ptr;
    double tmp = pp->size;

    if (partition) {
        partnum = find_partition(&(tmp), 0);
    }

    PPR_START(line, ptr);
    PPR_INT(ptr, pp->partkey);
    PPR_VSTR(ptr, pp->type,
              (columnar)?(long)P_TYPE_LEN:pp->tlen);
    PPR_INT(ptr, pp->size);
    PPR_STR(ptr, pp->brand, P_BRND_LEN);
    PPR_VSTR(ptr, pp->name,
              (columnar)?(long)P_NAME_LEN:pp->nlen);
    PPR_STR(ptr, pp->container, P_CNTR_LEN);
    PPR_STR(ptr, pp->mfg, P_MFG_LEN);
    PPR_MONEY(ptr, pp->retailprice);
    PPR_VSTR(ptr, pp->comment,
              (columnar)?(long)(ceil(P_CMNT_LEN * V_STR_HGH)):pp-
>clen);
    PPR_END(ptr);

    if (write(out[partnum], line, strlen(line)) == -1) {
        fprintf(stderr, "Error when writing parts data\n");
        Exitprog(errno);
    }

    return(0);
}

int
ld_psupp (part_t *pp, int mode)
{
    static int count = 0;

    if (! count++)
        printf("%s %s\n",
              "No load routine has been defined for the",
              "psupp table\n");
}

```

```

    return(0);
}

int
hd_time (FILE *f)
{
    static int count = 0;

    if (! count++)
        printf("No header has been defined for the time table\n");

    return(0);
}

int
ld_time (dss_time_t *tp, int mode)
{
    static int count = 0;

    if (! count++)
        printf("No load routine has been defined for the time table\n");

    return(0);
}

int
hd_supp (FILE *f)
{
    static int count = 0;

    if (! count++)
        printf("No header has been defined for the supplier table\n");

    return(0);
}

int
ld_supp (supplier_t *sp, int mode)
{
    static int count = 0;

    if (! count++)
        printf("%s %s\n",
            "No load routine has been defined",
            "for the supplier table\n");

    return(0);
}

int
hd_order (FILE *f)
{
    static int count = 0;

    if (! count++)
        printf("No header has been defined for the order table\n");

    return(0);
}

int
ld_order (order_t *o, int mode)
{

```

```

    char line[256];
    char *ptr;
    static int last_mode = 0;

    if (mode != last_mode)
        last_mode = mode;

    if (partition) {
        (void) strcpy(tmpdate, o->odate);
        partnum = find_partition(tmpdate, 2);
    }
    PPR_STRT(line, ptr);
    PPR_STR(ptr, o->odate, DATE_LEN);
    PPR_BCD2(ptr, o->okey_high, o->okey_low);
    PPR_INT(ptr, o->custkey);
    PPR_STR(ptr, o->opriority, O_OPRIO_LEN);
    PPR_INT(ptr, o->spriority);
    PPR_STR(ptr, o->clerk, O_CLRK_LEN);
    PPR_CHR(ptr, o->orderstatus);
    PPR_MONEY(ptr, o->totalprice);
    PPR_VSTR(ptr, o->comment,
        (columnar)?(long)(ceil(O_CMNT_LEN *
V_STR_HGH)):o->clen);
    PPR_END(ptr);

    if(write(out[partnum], line, strlen(line)) == -1) {
        fprintf(stderr, "Error when writing order data\n");
        Exitprog(errno);
    }

    return(0);
}

ld_line (order_t *o, int mode)
{
    char line[256];
    char *ptr;
    static int last_mode = 0;
    long i;

    if ( mode != last_mode)
        last_mode = mode;

    for (i = 0; i < o->lines; i++) {
        if (partition) {
            (void) strcpy(tmpdate, o->l[i].sdate);
            partnum = find_partition(tmpdate, 2);
        }
        PPR_STRT(line, ptr);
        PPR_STR(ptr, o->l[i].sdate, DATE_LEN);
        PPR_BCD2(ptr, o->l[i].okey_high, o->l[i].okey_low);
        PPR_MONEY(ptr, o->l[i].discount);
        PPR_MONEY(ptr, o->l[i].eprice);
        PPR_INT(ptr, o->l[i].suppkey);
        PPR_INT(ptr, o->l[i].quantity);
        PPR_CHR(ptr, o->l[i].rflag[0]);
        PPR_INT(ptr, o->l[i].partkey);
        PPR_CHR(ptr, o->l[i].lstatus[0]);
        PPR_MONEY(ptr, o->l[i].tax);
        PPR_STR(ptr, o->l[i].cdate, DATE_LEN);
        PPR_STR(ptr, o->l[i].rdate, DATE_LEN);
        PPR_STR(ptr, o->l[i].shipmode, L_SMODE_LEN);
        PPR_INT(ptr, o->l[i].lcnt);
        PPR_STR(ptr, o->l[i].shipinstruct, L_INST_LEN);
        PPR_VSTR(ptr, o->l[i].comment,
            (columnar)?(long)(ceil(L_CMNT_LEN * V_STR_HGH)):o-
>l[i].clen);
        PPR_END(ptr);
    }

```

```

        if(write(out[(partnum)], line, strlen(line)) == -1) {
            fprintf(stderr, "Error when writing lineitem data\n");
            Exitprog(errno);
        }
    }

    return(0);
}

int
hd_psupp (FILE *f)
{
    static int count = 0;

    if (! count++)
        printf("%s %s\n",
            "No header has been defined for the",
            "part supplier table");

    return(0);
}

int
hd_line (FILE *f)
{
    static int count = 0;

    if (! count++)
        printf("No header has been defined for the lineitem table\n");

    return(0);
}

int
hd_nation (FILE *f)
{
    static int count = 0;

    if (! count++)
        printf("No header has been defined for the nation table\n");

    return(0);
}

int
ld_nation (code_t *cp, int mode)
{
    static int count = 0;

    if (! count++)
        printf("%s %s\n",
            "No load routine has been defined",
            "for the nation table");

    return(0);
}

int
ld_region (code_t *cp, int mode)
{
    static int count = 0;

    if (! count++)
        printf("%s %s\n",
            "No load routine has been defined",
            "for the region table");

    return(0);
}

int
ld_order_line (order_t *p, int mode)
{
    ld_order(p, mode);
    ld_line (p, mode);

    return(0);
}

int
hd_order_line (FILE *f)
{
    hd_order(f);
    hd_line (f);

    return(0);
}

int
ld_part_psupp (part_t *p, int mode)
{
    ld_part(p, mode);
    ld_psupp (p, mode);

    return(0);
}

int
hd_part_psupp (FILE *f)
{
    hd_part(f);
    hd_psupp(f);

    return(0);
}

PART.C
/*=====
=====+
|      Copyright (c) 1997  Oracle Corp. Redwood Shores, CA      |
|      ADVANCED TECH PERFORMANCE GROUP                          |
|                                                                  |
|      All Rights Reserved                                       |
|                                                                  |
+=====
=====+
| FILENAME
|   part.c

```

```

| DESCRIPTION
| Functions to support partitioning.
| MODIFIED
| pswong 03/13/97 - created
| fjee 06/02/97 - add shared permission for open

+=====
=====*/

#include "part.h"

extern FILE *par;
extern int *out;
extern char delim;
char outname[FLEN];
extern int use_outparam;
FILE *outparam;
extern long numpart;
extern vals *values;

void process_param_file()
{

    char ofile[128];
    char line[128];
    char *pos, *str;
    int i,j;
    int retcode;
    struct stat fstats;

    /* get delimiter */

    fscanf(par, "%c\n", &delim);

    /* get number of partitions */

    fscanf(par, "%ld\n", &numpart);

    /* malloc some structures */

    values = (vals *) malloc(sizeof(vals));
    out = (int *) malloc(numpart * sizeof(int));

    /* extract the column types */

    fscanf(par, "%d\n", &(values->type));

    /* extract column position */

    fscanf(par, "%d\n", &(values->pos));

    /* setup output file descriptors */

    for (j=0; j<numpart; j++) {
        if (use_outparam) {
            /* get the file names */
            if (fscanf(outparam, "%s\n", ofile) == NULL) {
                fprintf(stderr, "Error: Number of entries in output parameter
file less than the number of partitions\n");
                (void) Exitprog(-1);
            }
        } else {
            sprintf(ofile, "%s.%d", (char *) outname, (j+1));
        }
        out[j] = open(ofile,
O_WRONLY|O_APPEND|O_CREAT,S_IWRITE|S_IREAD);
        if (out[j] == -1) {
            fprintf(stderr, "Unable to open file '%s'.\n", ofile);

            fprintf(stderr, "%s: %s\n", ofile, strerror(errno));
            (void) Exitprog(-1);
        }
    }

    /* allocate memory for the values */

    switch(values->type) {
    case NUM_TYP:
        values->vals = (double *) malloc(sizeof(double) * numpart);
        break;
    case STR_TYP:
        values->vals = (char **) malloc(sizeof(char *) * numpart);
        break;
    case DAT_TYP:
        values->vals = (dtyp *) malloc(sizeof(dtyp) * numpart);
        break;
    }

    /* now read the values */
    /* could have easily done by fscanf */

    for (j=0; j<numpart; j++) {
        if (fgets(line, 127, par) == NULL) {
            fprintf(stderr, "Error: number of boundaries definition does not
equal to the number of partitons.\n");
            Exitprog(-1);
        }
        str = strtok(line, DELIM);
        switch(values->type) {
        case NUM_TYP:
            ((double *)values->vals)[j] = (double) atof(str);
            break;
        case STR_TYP:
            ((char **)values->vals)[j] = (char *) malloc(strlen(str) + 1);
            strcpy(((char **)values->vals)[j], str);
            break;
        case DAT_TYP:
            ASS_DATE(((dtyp *)values->vals)[j], str);
            break;
        }
    }
}

void Exitprog(ern)
int ern;
{
    int i;

    /* free some memory */

    if (values->type == STR_TYP) {
        for(i=0; i<numpart; i++)
            free(((char **)values->vals)[i]);
    }

    for (i=0; i<numpart; i++)
        close(out[i]);

    free(values->vals);
    free(out);
    free(values);
    exit(ern);
}

```



```

/* if we are larger than the very last partition */
if (hint == (numpart - 1)) {
    if (fp(ptr_arith(arr, hint, type), item) <= 0)
        return (numpart-1);
    else {
        if (fp(ptr_arith(arr, (hint-1), type), item) <= 0)
            return hint;
        else {
            hint--;
            continue;
        }
    }
}

if (fp(ptr_arith(arr, hint, type), item) > 0) {
    /* let's see if item is in the current partition */
    if (fp(ptr_arith(arr, (hint-1), type), item) <= 0)
        return hint;
    else
        hint--;
} else hint++;
}
}

```

PART.H

```

/*=====
=====+
|   Copyright (c) 1997 Oracle Corp, Redwood Shores, CA   |
|   ADVANCED TECH PERFORMANCE GROUP                     |
|                                                         |
|   All Rights Reserved                                   |
|                                                         |
+=====
=====+
| FILENAME
|   part.h
| DESCRIPTION
|   Header file for dbgen partitioning extension.
| MODIFIED
|   pswong 03/13/97 - created
+=====
=====*/

#include <windows.h>
#include <winbase.h>
#include <io.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <share.h>

#define FLEN 128
#define BLEN 10240

typedef struct dt {
    int year;
    int month;
    int day;
} dtyp;

typedef struct val {
    int type;

```

```

    int pos;
    void *vals;
} vals;

typedef struct ot {
    int          optr;
    int          olen;
    char         obuf[BLEN];
    HANDLE       omutex;
} otype;

#define ASS_DATE(dat, str) \
{ dat.year = atoi(strtok(str, "-\n")); \
  dat.month = atoi(strtok(NULL, "-\n")); \
  dat.day = atoi(strtok(NULL, "-\n")); }

#define MINV(x,y) ((x < y) ? x : y)

/* some marco to do the print */

#ifndef SEPARATOR
#define SEPARATOR '|' /* field speparator for generated flat files */
#endif

#define PPR_STR(f, str, len) \
    if (columnar) sprintf(f, "%-s", len, str); \
    else sprintf(f, "%s%c", str, SEPARATOR); \
    f += strlen(f)

#define PPR_VSTR(f, str, len) PPR_STR(f, str, len)

#define PPR_INT(f, long) \
    if (columnar) sprintf(f, "%12d", long); \
    else sprintf(f, "%ld%c", long, SEPARATOR); \
    f += strlen(f)

#define PPR_BCD2(f, high, low) \
    if (high == 0) \
        if (columnar) sprintf(f, "%12d", low); \
        else sprintf(f, "%ld%c", low, SEPARATOR); \
    else \
        if (columnar) sprintf(f, "%5ld%07d", high, low); \
        else sprintf(f, "%ld%07d%c", high, low, SEPARATOR); \
    f += strlen(f)

#define PPR_KEY(f, long) \
    sprintf(f, "%ld", long); \
    f += strlen(f)

#define MONEY_COL_FMT "%12ld.%02ld"
#define PPR_MONEY(fp, flt) \
    if (columnar) sprintf(fp, MONEY_COL_FMT, \
        flt/100, ((flt < 0)?-flt:flt)%100); \
    else sprintf(fp, "%ld.%02ld%c", \
        flt/100, ((flt < 0)?-flt:flt)%100, SEPARATOR); \
    fp += strlen(fp)

#define PPR_CHR(fp, chr) \
    if (columnar) sprintf(fp, "%c ", chr); \
    else sprintf(fp, "%c%c", chr, SEPARATOR); \
    fp += strlen(fp)

#define PPR_STRT(dest, ptr) (ptr = dest)
#define PPR_END(fp)    sprintf(fp, "\n") /* finish the record here */

#ifndef MDY_DATE
#define PR_DATE(tgt, yr, mn, dy) \
    sprintf(tgt, "%02d-%02d-19%02d", mn, dy, yr)

```

```
#else
#define PR_DATE(tgt, yr, mn, dy)    \
    sprintf(tgt, "19%02d-%02d-%02d", yr, mn, dy)
#endif /* DATE_FORMAT */

/* Could add a integer type and a character type to allow faster
processing. */
#define NUM_TYP 0    /* number type */
#define STR_TYP 1    /* str/char type */
#define DAT_TYP 2    /* date type */

#define DELIM "\n"

void Exitprog();
void process_param_file();
int flcmp();
int datcmp();
int strgcmp();
int part_search();
int find_partition();
```

Appendix H: Price Quote

From: Ray Glasstone [RGLASSTO@us.oracle.com]
Sent: Thursday, June 18, 1998 7:24 PM
To: Nikolaiev, Mike
Cc: Harris, Susan
Subject: Price quote for your Xeon TPC-D benchmark.
Hi Mike,

In response to your inquiry, here is the U.S. pricing and availability information for the specified configuration (4 CPU Xeon):

Oracle8 License: \$ 50,160

Oracle Silver support for five years: \$ 66,640

Oracle8 version 8.0.4 availability date: now

Regards,

Ray.

ALLSTAR SYSTEMS*

Item Description	Stock Number	Price Each	QTY	Total	5 Year Service	Total Price
ProLiant 7000-6/400-1M	312150-001	\$25,800.00	1	\$25,800.00		\$25,800.00
ProLiant 7000 Maintenance Upgrade	276535-000	\$1,075.00	1		\$1,075.00	\$1,075.00
Enhanced Keyboard	110562-001	\$76.00	1	\$76.00		\$76.00
42U 19" Rack	165753-001	\$1,712.13	1	\$1,712.13		\$1,712.13
Rack Sidewall Kit (For 42U Rack)	165652-001	\$212.32	1	\$212.32		\$212.32
ProLiant Xeon 400 - 1MB Processor Kit	312286-B21	\$4,835.00	3	\$14,835.00		\$14,835.00
1 GB DIMM Kit	241774-B21	\$6,257.94	4	\$25,031.76		\$25,031.76
Fibre Channel Host Adapter Kit / PCI	223180-B21	\$1,788.59	1	\$1,788.59		\$1,788.59
Compaq VGA V50 - 15" Color Monitor	254150-001	\$267.93	1	\$267.93		\$267.93
24 X 7 -4HR Server Maintenance Upgrade					\$11,308.47	\$11,308.47
Compaq Uninterruptible Power Supply T2	242688-005	\$954.23	1	\$954.23		\$954.23
Fibre Channel Storage Hub 7	234453-001	\$1,112.89	1	\$1,112.89		\$1,112.89
Fibre Channel Encl. & Array Ctrlr Kit, Ra	223100-001	\$6,882.77	6	\$41,296.62		\$41,296.62
ProLiant Storage Maintenance Upgrade	276537-001	\$430.00	6		\$2,580.00	\$2,580.00
24 X 7 -4HR Storage Maintenance Upgrade					\$7,028.36	\$7,028.36
4.3 GB Pluggable SCSI-2 Drive	146742-005	\$751.43	1	\$751.43		\$751.43
8.1 GB 10k rpm Pluggable SCSI-3 Drive	336357-B21	\$1,395.43	84	\$117,298.70		\$117,298.70
48GB DAT Drive	295353-B21	\$766.29	1	\$766.29		\$766.29
MKS Toolkit for NT		\$499.00	1	\$499.00		\$499.00
Microsoft Visual C++ and Subscription Service		\$499.00	1	\$499.00	\$2,495.00	\$2,994.00
Microsoft Windows NT Enterprise v. 4.0		\$3,999.00	1	\$3,999.00		\$3,999.00
Totals:				\$236,901.88	\$24,486.83	\$261,388.71
Prices include a large volume discount and may vary when items are purchased separately.						
Service prices based upon a service contract for all items listed.						
The above quotation is valid for 60 days.						

Houston Office: 6401 Southwest Freeway, Houston, Texas 77074, (713) 795-2000 • Dallas Office: 14202 Proton Road, Dallas, Texas 75244, (214) 774-7000
Austin Office: 4030 West Braker Lane, Suite 305, Austin, Texas 78759, (512) 231-8800 • El Paso Office: 1855 Trawood, Suite #209, El Paso, Texas 79935, (915) 590-6589
McAllen Office: 709 East Esperanza, Suite #E, McAllen, Texas 78501, (956) 928-1020