



Exam : 1Z1-054

Title : Oracle Database 11g:
Performance Tuning

Version : Demo

1. After running SQL Performance Analyzer (SPA), you observe a few regressed SQL statements in the SPA output. Identify the two actions that you would suggest for these regressed SQL statements.

(Choose two.)

- A. Running SQL Access Advisor
- B. Adding them to SQL Plan Baseline
- C. Submitting them to SQL Tuning Advisor
- D. Running Automatic Database Diagnostic Monitor (ADDM)

ANSWER: BC

2. View the Exhibit exhibit1 to examine the series of SQL commands and parameter settings.

```
SQL> SHOW PARAMETER OPTIMIZER
```

NAME	TYPE	VALUE
optimizer_capture_sql_plan_baselines	boolean	TRUE
optimizer_dynamic_sampling	integer	2
optimizer_features_enable	string	11.1.0.6
optimizer_index_caching	integer	0
optimizer_index_cost_adj	integer	100
optimizer_mode	string	ALL_ROWS
optimizer_secure_view_merging	boolean	TRUE
optimizer_use_invisible_indexes	boolean	FALSE
optimizer_use_pending_statistics	boolean	FALSE
optimizer_use_sql_plan_baselines	boolean	TRUE

```
SQL> SELECT * FROM sh.sales WHERE quantity_sold > 40 ORDER BY prod_id;
SQL> SELECT * FROM sh.sales WHERE quantity_sold > 40 ORDER BY prod_id;
SQL> ALTER SESSION SET OPTIMIZER_MODE=FIRST_ROWS;
SQL> SELECT * FROM sh.sales WHERE quantity_sold > 40 ORDER BY prod_id;
```

View the Exhibit exhibit2 to examine the plans available in the SQL plan baseline.

Select	Name	SQL Text	Enabled	Accepted	Fixed	Auto Purge	Created	Last Modified
<input type="checkbox"/>	SYS_SQL_PLAN_89447021cf314e9e	select * from hr.employees where job_id='CLERK'	YES	YES	NO	YES	Jul 20, 2008 7:02:30 PM	Jul 20, 2008 7:16:48 PM
<input type="checkbox"/>	SYS_SQL_PLAN_894470210572d2e8	select * from hr.employees where job_id='CLERK'	YES	NO	NO	YES	Jul 20, 2008 7:20:45 PM	Jul 20, 2008 7:20:45 PM
<input type="checkbox"/>	SYS_SQL_PLAN_7ed8568135b3cdca	SELECT NAME NAME COL PLUS SHOW PARAM,DECODE (TYPE,1...	YES	YES	NO	YES	Jul 21, 2008 2:40:44 PM	Jul 21, 2008 2:40:44 PM
<input type="checkbox"/>	SYS_SQL_PLAN_4698b35ddf463620	select * from table(dbms_xplan.display (null,null,'...	YES	YES	NO	YES	Jul 20, 2008 7:04:22 PM	Jul 20, 2008 7:04:22 PM
<input type="checkbox"/>	SYS_SQL_PLAN_467a776254bc8843	select * from sh.sales where quantity_sold > 40 or...	YES	YES	NO	YES	Jul 21, 2008 2:25:42 PM	Jul 21, 2008 2:25:42 PM
<input type="checkbox"/>	SYS_SQL_PLAN_467a776211df68d0	select * from sh.sales where quantity_sold > 40 or...	YES	NO	YES	YES	Jul 21, 2008 2:41:22 PM	Jul 21, 2008 2:41:56 PM

The first plan (in red) is created when OPTIMIZER_MODE is set to ALL_ROWS and the second plan (in blue) is created when OPTIMIZER_MODE is set to FIRST_ROWS.

Which SQL plan baseline would be used if the SQL query in exhibit1 is executed again when the value of OPTIMIZER_MODE is set to FIRST_ROWS?

- A. the second plan, because it is a fixed plan
- B. the first plan, because it is an accepted plan
- C. the second plan, because it is the latest generated plan in FIRST_ROW mode
- D. A new plan, because the second plan in FIRST_ROW mode is not an accepted plan

ANSWER: B

3. You work as a DBA for a company and you have the responsibility of managing one of its online transaction processing (OLTP) systems. The database encountered performance-related problems and you generated an Automatic Workload Repository (AWR) report to investigate it further. View the Exhibits and examine the AWR report.

Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
DB CPU		584		29.08	
library cache: mutex X	14,721	71	5	3.53	Concurrency
latch: shared pool	1,158	55	48	2.76	Concurrency
cursor: pin S wait on X	3,777	50	13	2.50	Concurrency
log file sync	672	17	25	0.83	Commit

Time Model Statistics

- Total time in database user-calls (DB Time): 2008.5s
- Statistics including the word "background" measure background process time, and so do not contribute to the DB time statistic
- Ordered by % of DB time desc, Statistic name

Statistic Name	Time (s)	% of DB Time
sql execute elapsed time	1,731.94	86.23
DB CPU	584.11	29.08
parse time elapsed	533.72	26.57
hard parse elapsed time	416.43	20.73
connection management call elapsed time	33.26	1.66
PL/SQL compilation elapsed time	10.58	0.53
Java execution elapsed time	8.01	0.40
failed parse elapsed time	5.20	0.26
PL/SQL execution elapsed time	3.66	0.18
hard parse (sharing criteria) elapsed time	1.94	0.10
hard parse (bind mismatch) elapsed time	1.33	0.07
sequence load elapsed time	0.41	0.02
repeated bind elapsed time	0.05	0.00
DB time	2,008.48	
background elapsed time	32.06	
background cpu time	4.79	

Load Profile

	Per Second	Per Transaction	Per Exec	Per Call
DB Time(s):	3.8	12.6	0.01	0.00
DB CPU(s):	1.1	3.7	0.00	0.00
Redo size:	6,062.3	20,190.1		
Logical reads:	5,982.5	19,924.3		
Block changes:	25.5	84.9		
Physical reads:	2,778.2	9,252.7		
Physical writes:	2.9	9.7		
User calls:	1,263.4	4,207.7		
Parses:	506.6	1,687.3		
Hard parses:	53.3	177.5		
W/A MB processed:	726,646.9	2,420,040.5		
Logons:	1.1	3.5		
Executes:	513.1	1,708.9		
Rollbacks:	0.1	0.3		
Transactions:	0.3			

Dictionary Cache Stats

- "Pot Misses" should be very low (< 2% in most cases)
- "Final Usage" is the number of cache entries being used

Cache	Get Requests	Pct Miss	Scan Reqs	Pct Miss	Mod Reqs	Final Usage
do_awr_control	13	69.23	0		2	1
do_database_links	1,074	0.56	0		0	0
do_global_objs	15,419	2.87	0		0	13
do_histogram_data	77,585	21.21	0		0	571
do_histogram_defs	168,045	23.16	0		0	1,014
do_object_grants	44,042	4.17	0		0	59
do_objects	358,789	3.30	0		0	398
do_profiles	548	2.19	0		0	1
do_rollback_segments	230	0.00	0		0	38
do_segments	99,805	15.72	0		5	279
do_sequences	25	100.00	0		25	0
do_tablespaces	85,888	0.04	0		0	5
do_users	179,387	0.35	0		0	20
global database name	927	0.11	0		0	1
kqsubheap_object	197	30.46	0		0	0
outstanding_alerts	19	94.74	0		0	1

[Back to Top](#)

Library Cache Activity

- "Pot Misses" should be very low

Namespace	Get Requests	Pct Miss	Pin Requests	Pct Miss	Reloads	Invali- dations
BODY	1,832	1.36	3,673	1.55	23	0
CLUSTER	2,761	1.81	1,590	3.14	0	0
INDEX	947	35.59	947	35.90	1	0
JAVA DATA	4	75.00	873	0.89	0	0
SQL AREA	340,330	23.79	602,683	12.78	22,142	5,231
TABLE/PROCEDURE	145,469	2.49	191,059	8.55	5,812	0
TRIGGER	5,539	0.23	5,539	0.29	0	0

What could be the problem in this database?

- Java pool is not configured.
- The CPU in the system is slow.
- The shared pool size is inadequate.
- The database buffer cache is inadequate.
- The OPEN_CURSORS parameter is set to a small value.

ANSWER: C

4. You are working on an online transaction processing (OLTP) system. You detected free buffer waits events for your database instance running in a machine that has multiple CPUs. You increased the database buffer cache size as the first step. After a few hours of work on the database, further investigation shows that the same event is being recorded.

What would be your next step to avoid this event in future?

- A. Decrease the value of the DBWR_IO_SLAVES parameter.
- B. Set the USE_INDIRECT_DATA_BUFFERS parameter to TRUE.
- C. Increase the value of the DB_WRITER_PROCESSES parameter.
- D. Increase the value of the DB_FILE_MULTIBLOCK_READ_COUNT parameter.

ANSWER: C

5. You are working on a development database that was upgraded to Oracle Database 11g from Oracle Database 9i. An ADDM finding in this database says that the shared pool is inadequately sized, as shown in the Exhibit.

ORACLE Enterprise Manager 11g Database Control

Database Instance: [ord.us.oracle.com](#) > [Advisor Central](#) >
 Automatic Database Diagnostic Monitor (ADDM):SYS.ADDM:1159023676_1_1390 >

Performance Finding Details: Undersized Shared Pool

Finding: **Hard parses due to an inadequately sized shared pool were consuming significant database time.** [Finding History](#)

Impact (Active Sessions): **.69**
 Impact (%): **42.5**

Period Start Time: **Jul 29, 2008 7:23:34 PM GMT+07:00**
 Period Duration (minutes): **24.4**
 Filtered: **No** [Filters](#)

You diagnosed that this is due to different kinds of workloads and this occurs only during peak hours. You tried to resize this by shrinking the database buffer cache but that caused inadequate buffer cache problems. The following are the related parameter settings:

```
SQL> show parameter sga
NAME TYPE VALUE
-----
lock_sga boolean FALSE
pre_page_sga boolean FALSE
sga_max_size big integer 300M
sga_target big integer 0
SQL> show parameter target
NAME TYPE VALUE
-----
.....
fast_start_mttr_target integer 0
memory_max_target big integer 0
memory_target big integer 0
pga_aggregate_target big integer 100M
sga_target big integer 0
```

You want to balance the memory between the System Global Area (SGA) components within SGA without affecting the size of the Program Global Area (PGA).

Which action would solve this problem?

- A. Set the SGA_TARGET parameter to 300M.
- B. Set the SGA_MAX_SIZE parameter to 400M.
- C. Set the MEMORY_TARGET parameter to 100M.
- D. Set the MEMORY_MAX_TARGET parameter to 300M.

ANSWER: A

6. You observed that some of the queries are performing poorly on the SALES_RECORDS table in your database.

On further investigation, you find that at the end of each day the contents of the SALES_RECORDS table are transferred to the SALES table and deleted from the SALES_RECORDS table. The deleted operations cause the table to be sparsely populated.

You decided to use the ALTER TABLE...SHRINK SPACE COMPACT command to shrink the table.

Why would you choose this method? (Choose all that apply.)

- A. because it can be used during peak hours
- B. because it avoids unnecessary cursor invalidation
- C. because it adjusts the high-water mark (HWM) immediately
- D. because you have long-running queries that might span the shrink operation
- E. because it does not allow any data manipulation language (DML) operations, thereby making the shrink operation faster

ANSWER: ABD

7. View the Exhibit and examine a portion of the output obtained from the following query:

STAT_ID	STAT_NAME	VALUE
3649082374	DB time	61021783
2748282437	DB CPU	3890625
4157170894	background elapsed time	42472524
2451517896	background cpu time	2796875
4127043053	sequence load elapsed time	0
1431595225	parse time elapsed	10983653
372226525	hard parse elapsed time	10480831
2821698184	sql execute elapsed time	50353110
1990024365	connection management call elapsed time	855906

SQL> SELECT * FROM v\$sqlsys_time_model;

Select three correct interpretations of the time model statistics. (Choose three.)

- A. DB time includes the wait time of all the nonidle and idle user sessions.
- B. SQL execute elapsed time includes the time spent in performing fetches of query results.
- C. DB CPU includes the CPU time spent on database user-level calls and background CPU time.
- D. SQL execute elapsed time includes components of the hard parse elapsed time like bind elapsed time.
- E. DB time includes the connection management call elapsed time excluding the background process time.

ANSWER: BDE

8. A batch workload that historically completed in the maintenance window between 10:00 PM and midnight is currently showing poor performance and completing at 2 AM.

To help in the diagnosis of the performance degradation, the senior DBA in your organization asks you to execute the awrddrpt.sql script to produce a Compare Periods report.

Which two statements are true about the report produced by this script? (Choose two.)

- A. It is refreshed automatically based on the moving window baseline.
- B. It compares details between any two selected time periods of the same duration.
- C. It normalizes the statistics by the amount of time spent on the database for each time period.
- D. It compares details between two consecutive time periods of the same or different durations and is refreshed every 60 minutes.

ANSWER: BC

9. You plan to use adaptive thresholds as part of the performance tuning activity. You decide to increase the window size of the default moving window baseline for all metric observations and comparisons in your database. The following error occurs when you try to increase the window size through Enterprise Manager:

Failed to commit: ORA-13541: system moving window baseline size (1296000) greater than retention (1036800) ORA-06512: at "SYS.DBMS_WORKLOAD_REPOSITORY", line 601 ORA-06512: at line 2

Which action would allow you to perform the preceding task successfully?

- A. increasing the flashback retention period
- B. increasing the retention period for SQL Management Base
- C. increasing the undo retention period for the database instance
- D. increasing the Automatic Workload Repository (AWR) retention period

ANSWER: D

10. Identify two correct statements about the Active Session History (ASH) data. (Choose two.)

- A. A part of SGA memory is used to store ASH data as rolling buffer.
- B. The ASH data can be analyzed between any two small time intervals.
- C. All ASH data in memory are flushed to disk by MMON in every 3 seconds.
- D. All ASH data in memory are flushed to disk by MMNL process whenever the buffer is full.

ANSWER: AB

11. A user in a session executed the following SQL statement to set the optimizer mode:

```
ALTER SESSION SET OPTIMIZER_MODE = ALL_ROWS
```

What impact would it have on the goal of the optimizer for that session? (Choose all that apply.)

- A. Statement level OPTIMIZER_MODE hints take precedence over the session-level setting.
- B. The OPTIMIZER_MODE parameter set at instance level takes precedence over the session-level value.
- C. The optimizer uses a cost-based approach, regardless of the presence of statistics; it optimizes with a goal of best response time.
- D. The optimizer uses a cost-based approach for all SQL statements in the session, regardless of the presence of statistics; it optimizes with a goal of best throughput.

ANSWER: AD

12. You are working on an online transaction processing (OLTP) system. By day most of the application users perform queries accessing the most recently added or modified rows. The applications have most of the queries based on multiple tables. But at night, some batch processing is also done.

Which two actions would you recommend to choose a goal for the optimizer based on the needs of your application? (Choose two.)

- A.setting the OPTIMIZER_MODE parameter to ALL_ROWS at the instance level
- B.setting the OPTIMIZER_MODE parameter to FIRST_ROWS_n at the instance level
- C.asking the developer to add a hint /*ALL_ROWS*/ in the long-running batch processing queries
- D.asking the developer to add a hint /*FIRST_ROWS_n*/ in the long-running batch processing queries

ANSWER: BC

13. Which three factors influence the optimizer's behavior while choosing an optimization approach and goal for a SQL statement? (Choose three.)

- A.parsing of a SQL statement
- B.operating system (OS) statistics
- C.object statistics in the data dictionary
- D.the OPTIMIZER_MODE initialization parameter
- E.optimizer SQL hints for changing the query optimizer goal

ANSWER: CDE

14. Examine the initialization parameter values for the instance given below:

```
NAME TYPE VALUE
```

```
-----
```

```
optimizer_capture_sql_plan_baselines boolean FALSE
optimizer_dynamic_sampling integer 2
optimizer_features_enable string 11.1.0.6
optimizer_index_caching integer 0
optimizer_index_cost_adj integer 100
optimizer_mode string ALL_ROWS
db_file_multiblock_read_count integer 64
```

The index created on the column used in the WHERE clause of the query. You notice that the query is not using the index. Instead of an index scan, a full table scan is used.

View the Exhibit and examine the autotrace output for a query.

```
select * from employees where employee_id=107;
```

Execution Plan

```
-----
Plan hash value: 1601196873
```

```
-----
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	71	3 (0)	00:00:01
* 1	TABLE ACCESS FULL	T	1	71	3 (0)	00:00:01

```
-----
```

Predicate Information (identified by operation id):

```
-----
1 - filter("EMPLOYEE_ID"=107)
```

What could be the reason for it? (Choose all that apply.)

- A. The OPTIMIZER_INDEX_COST_ADJ initialization parameter has a low value.
- B. The DB_FILE_MULTIBLOCK_READ_COUNT initialization parameter has a low value.
- C. The statistics for the table and all the indexes associated with the table are not current.
- D. The table has less than DB_FILE_MULTIBLOCK_READ_COUNT blocks under the high-water mark.

ANSWER: CD

15. Examine the initialization parameter values for the instance given below:

NAME TYPE VALUE

```
-----
optimizer_capture_sql_plan_baselines boolean FALSE
optimizer_dynamic_sampling integer 2
optimizer_features_enable string 11.1.0.6
optimizer_index_caching integer 0
optimizer_index_cost_adj integer 100
```

optimizer_mode string ALL_ROWS

db_file_multiblock_read_count integer 64

You notice that the one of the queries is using a full table scan (view Exhibit1) instead of index unique scan (view Exhibit2). The index is present on the column that is accessed in the WHERE clause of the query. The cost for a full table scan is more than that for an index unique scan.

Why would the optimizer choose full table scan over index unique scan? (Choose all that apply.)

- A. The OPTIMIZER_INDEX_COST_ADJ initialization parameter is set to a low value.
- B. The OPTIMIZER_INDEX_COST_ADJ initialization parameter is set to a high value.
- C. The DB_FILE_MULTIBLOCK_READ_COUNT initialization parameter is set to a low value.
- D. The statistics for the table and all the indexes associated with the table are not current.

ANSWER: BD

16. The columns CUST_CITY, CUST_STATE_PROVINCE, and COUNTRY_ID are frequently used together in the WHERE clause of the queries. The CUSTOMERS table is a big table with 20 GB of data. You notice that the selectivity for these three columns varies from the selectivity that the optimizer calculates.

What would you recommend to influence the selectivity calculated by the optimizer?

- A. creating function-based indexes by concatenating all the columns
- B. updating histogram statistics for these columns by using the DBMS_STATS.GATHER_TABLE_STATS procedure
- C. using the DBMS_STATS.CREATE_EXTENDED_STATS function to create a virtual column and create index on the virtual column
- D. using the DBMS_STATS.CREATE_EXTENDED_STATS function to create a virtual column and the DBMS_STATS.GATHER_TABLE_STATS procedure to collect statistics on the virtual column

ANSWER: D

17. View Exhibit1 to examine the description of the CUSTOMERS table.

```
SQL> DESC customers
Name                               Null?    Type
-----
CUST_ID                             NOT NULL NUMBER
CUST_FIRST_NAME                     NOT NULL VARCHAR2 (20)
CUST_LAST_NAME                      NOT NULL VARCHAR2 (40)
CUST_GENDER                         NOT NULL CHAR (1)
CUST_YEAR_OF_BIRTH                 NOT NULL NUMBER (4)
CUST_MARITAL_STATUS                 VARCHAR2 (20)
CUST_STREET_ADDRESS                NOT NULL VARCHAR2 (40)
CUST_POSTAL_CODE                   NOT NULL VARCHAR2 (10)
CUST_CITY                          NOT NULL VARCHAR2 (30)
CUST_CITY_ID                       NOT NULL NUMBER
CUST_STATE_PROVINCE                NOT NULL VARCHAR2 (40)
CUST_STATE_PROVINCE_ID             NOT NULL NUMBER
COUNTRY_ID                         NOT NULL NUMBER
CUST_MAIN_PHONE_NUMBER             NOT NULL VARCHAR2 (25)
CUST_INCOME_LEVEL                  VARCHAR2 (30)
CUST_CREDIT_LIMIT                  NUMBER
CUST_EMAIL                         VARCHAR2 (30)
CUST_TOTAL                         NOT NULL VARCHAR2 (14)
CUST_TOTAL_ID                     NOT NULL NUMBER
CUST_SRC_ID                        NUMBER
CUST_EFF_FROM                      DATE
CUST_EFF_TO                        DATE
CUST_VALID                         VARCHAR2 (1)
```

You observed that optimizer selectivity is not accurate when the CUST_STATE_PROVINCE and COUNTRY_ID columns are used together in the WHERE clause of a query.

View Exhibit2 to examine the query execution plan and the commands executed to gather the statistics.

```

SQL> SELECT COUNT(*) FROM customers WHERE country_id=52790 and cust_state_province='CA';

COUNT(*)
-----
      3341

SQL> exec dbms_stats.gather_table_stats(null,'CUSTOMERS',method_opt=>'for all columns size 1');

PL/SQL procedure successfully completed.

SQL> explain plan for select count(*) from customers where COUNTRY_ID=52790 and CUST_STATE_PROVINCE='CA';

Explained.

SQL> select * from table(dbms_xplan.display())
      2 ;

PLAN_TABLE_OUTPUT
-----
Plan hash value: 296924608

-----
| Id | Operation          | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
-----
|  0 | SELECT STATEMENT   |               |      1 |    16 |    406 (1)| 00:00:05 |
|  1 |  SORT AGGREGATE    |               |      1 |    16 |             |          |
|* 2 |   TABLE ACCESS FULL| CUSTOMERS     |     20 |   320 |    406 (1)| 00:00:05 |
-----

Predicate Information (identified by operation id):

PLAN_TABLE_OUTPUT
-----

      2 - filter("CUST_STATE_PROVINCE"='CA' AND "COUNTRY_ID"=52790)

14 rows selected.

```

The optimizer predicts that 20 rows will be processed rather than the 3,341 rows, which is the actual number of rows returned from the table.

What can you do to make the optimizer detect the actual number of rows?

- A. Set the STATISTICS_LEVEL parameter to ALL.
- B. Set the OPTIMIZER_USE_PENDING_STATISTICS parameter to FALSE.
- C. Create extended statistics for the CUST_STATE_PROVINCE and COUNTRY_ID columns.
- D. Increase the STALE_PERCENT value for the CUSTOMERS table by using the DBMS_STATS.SET_TABLE_PREFS procedure.

ANSWER: C

18. View Exhibit1 to examine the description of the CUSTOMERS table.

```

SQL> DESC customers
Name                               Null?    Type
-----
CUST_ID                             NOT NULL NUMBER
CUST_FIRST_NAME                     NOT NULL VARCHAR2 (20)
CUST_LAST_NAME                      NOT NULL VARCHAR2 (40)
CUST_GENDER                         NOT NULL CHAR (1)
CUST_YEAR_OF_BIRTH                  NOT NULL NUMBER (4)
CUST_MARITAL_STATUS                 VARCHAR2 (20)
CUST_STREET_ADDRESS                 NOT NULL VARCHAR2 (40)
CUST_POSTAL_CODE                    NOT NULL VARCHAR2 (10)
CUST_CITY                           NOT NULL VARCHAR2 (30)
CUST_CITY_ID                        NOT NULL NUMBER
CUST_STATE_PROVINCE                 NOT NULL VARCHAR2 (40)
CUST_STATE_PROVINCE_ID              NOT NULL NUMBER
COUNTRY_ID                          NOT NULL NUMBER
CUST_MAIN_PHONE_NUMBER              NOT NULL VARCHAR2 (25)
CUST_INCOME_LEVEL                   VARCHAR2 (30)
CUST_CREDIT_LIMIT                   NUMBER
CUST_EMAIL                          VARCHAR2 (30)
CUST_TOTAL                          NOT NULL VARCHAR2 (14)
CUST_TOTAL_ID                       NOT NULL NUMBER
CUST_SRC_ID                         NUMBER
CUST_EFF_FROM                       DATE
CUST_EFF_TO                         DATE
CUST_VALID                          VARCHAR2 (1)

```

The CUSTOMERS table has been updated heavily today. In a frequently used SQL statement, you notice that estimated rows and the actual number of rows fetched differ greatly. The COUNTRY_ID column has an index.

View Exhibit2 and examine the query execution plan.

```
SQL> SELECT cust_id, cust_last_name, cust_total
2 FROM customers
3 WHERE country_id = 52790
```

296320 rows selected.

Execution Plan

Plan hash value: 2008213504

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		73966	2383K	798 (1)	00:00:10
* 1	TABLE ACCESS FULL	CUSTOMERS	73966	2383K	798 (1)	00:00:10

Predicate Information (identified by operation id):

1 - filter("COUNTRY_ID"=52790)

Statistics

1 recursive calls
0 db block gets
27844 consistent gets
8597 physical reads
0 redo size
6373693 bytes sent via SQL*Net to client
217714 bytes received via SQL*Net from client
19756 SQL*Net roundtrips to/from client
0 sorts (memory)
0 sorts (disk)
296320 rows processed

What would you recommend to improve the optimizer's estimation?

- A. setting the STATISTICS_LEVEL parameter to ALL
- B. setting the OPTIMIZER_USE_PENDING_STATISTICS parameter to FALSE
- C. creating extended statistics for the CUST_LAST_NAME, CUST_ID, and CUST_TOTAL columns
- D. updating the statistics for the CUSTOMERS table by using the DBMS_STATS.GATHER_TABLE_STATS procedure

ANSWER: D

19. You are working on a decision support system (DSS). The index is available on the COUNTRY_ID column of the CUSTOMERS table.

View the Exhibit and examine the parameter settings and the query execution plan.

NAME	TYPE	VALUE
db_file_multiblock_read_count	integer	49

```
SQL> SELECT BLOCKS, EMPTY_BLOCKS FROM ALL_TABLES
WHERE TABLE_NAME = 'CUST';
```

BLOCKS	EMPTY_BLOCKS
2902	0

```
SQL> EXEC DBMS_STATS.GATHER_TABLE_STATS('SH','CUST');
```

PL/SQL procedure successfully completed.

```
SQL> select cust_id, cust_last_name from sh.cust where country_id=52790;
```

74080 rows selected.

Execution Plan

Plan hash value: 260468903

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		74150	1303K	793 (1)	00:00:10
* 1	TABLE ACCESS FULL	CUST	74150	1303K	793 (1)	00:00:10

Predicate Information (identified by operation id):

1 - filter("COUNTRY_ID"=52790)

Statistics

0	recursive calls
0	db block gets
7659	consistent gets
2848	physical reads
0	redo size
1584703	bytes sent via SQL*Net to client
54738	bytes received via SQL*Net from client
4940	SQL*Net roundtrips to/from client

Why is the query using a full table scan instead of an index scan?

- A. because the histogram statistics for the COUNTRY_ID column are not updated
- B. because the index statistics for the index on the COUNTRY_ID column are not current
- C. because the DB_FILE_MULTIBLOCK_READ_COUNT initialization parameter is set to a higher value
- D. because the optimizer predicts that most of the blocks in the table are accessed. Therefore, it uses a full table scan, even though indexes are available.

ANSWER: D

20. View Exhibit1 and examine the indexes on the CUSTOMERS table.

```
SQL> SELECT i.table_name, column_name, o.object_name , o.object_id, i.index_type
FROM   user_objects o,
       user_indexes i,
       user_ind_columns c
WHERE  o.object_type= 'INDEX'
AND    i.table_name LIKE '%CUST%'
AND    i.index_name = o.object_name
AND    i.index_name = c.index_name
```

TABLE_NAME	COLUMN_NAME	OBJECT_NAME	OBJECT_ID	INDEX_TYPE
CUSTOMERS	CUST_GENDER	CUSTOMERS_GENDER_BIX	70685	BITMAP
CUSTOMERS	CUST_MARITAL_STATUS	CUSTOMERS_MARITAL_BIX	70686	BITMAP
CUSTOMERS	CUST_ID	CUSTOMERS_PK	70473	NORMAL
CUSTOMERS	CUST_YEAR_OF_BIRTH	CUSTOMERS_YOB_BIX	70687	BITMAP
CUSTOMERS	COUNTRY_ID	CUST_COUNTRY	79047	NORMAL

The statistics for the CUSTOMERS table have been updated recently by using the following command:

```
SQL> EXEC DBMS_STATS.GATHER_TABLE_STATS('SH','CUSTOMERS',method_opt=>'FOR ALL
INDEXED COLUMNS SIZE AUTO');
```

View Exhibit2 to examine a query plan. Even though the index is present on the COUNTRY_ID and CUST_GENDER columns, the query uses a full table scan. What could be the reason?

```
SQL> SELECT i.table_name, column_name, o.object_name , o.object_id, i.index_type
FROM   user_objects o,
       user_indexes i,
       user_ind_columns c
WHERE  o.object_type= 'INDEX'
AND    i.table_name LIKE '%CUST%'
AND    i.index_name = o.object_name
AND    i.index_name = c.index_name
```

TABLE_NAME	COLUMN_NAME	OBJECT_NAME	OBJECT_ID	INDEX_TYPE
CUSTOMERS	CUST_GENDER	CUSTOMERS_GENDER_BIX	70685	BITMAP
CUSTOMERS	CUST_MARITAL_STATUS	CUSTOMERS_MARITAL_BIX	70686	BITMAP
CUSTOMERS	CUST_ID	CUSTOMERS_PK	70473	NORMAL
CUSTOMERS	CUST_YEAR_OF_BIRTH	CUSTOMERS_YOB_BIX	70687	BITMAP
CUSTOMERS	COUNTRY_ID	CUST_COUNTRY	79047	NORMAL

- A. because the histogram statistics for the COUNTRY_ID column are not updated
- B. because the DB_FILE_MULTIBLOCK_READ_COUNT initialization parameter is set to a high value
- C. because the optimizer calculates the cost of accessing blocks by using a full table scan to be less as compared to index scans, even though indexes are available
- D. because indexes on CUST_GENDER and COUNTRY_ID columns are of different types, the index on the CUST_GENDER column is bitmap index, and on COUNTRY_ID columns is btree index.

ANSWER: C