1Z0-117$^{Q\&As}$

Oracle Database 11g Release 2: SQL Tuning Exam

Pass home 1Z0-117 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

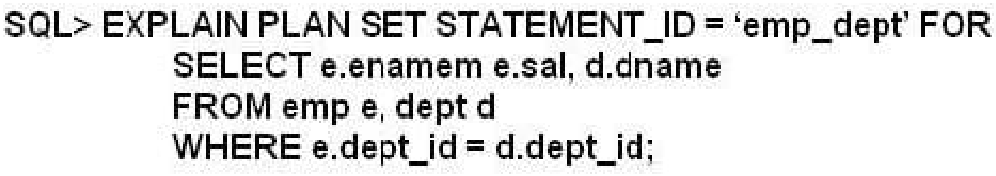
https://www.leads4pass.com/1z0-117.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by home Official Exam Center

QUESTION 1

You executed the following statement:



```
SQL> EXPLAIN PLAN SET STATEMENT_ID = ‘emp_dept’ FOR
          SELECT e.enamem e.sal, d.dname
          FROM emp e, dept d
          WHERE e.dept_id = d.dept_id;
```

Which three statements are true about EXPLAIN PLAN?

A. The execution plan is saved in PLAN_TABLE without executing the query.

B. The execution plan for the query is generated and displayed immediately as the output.

C. The execution plan generated may not necessarily be the execution plan used during query execution.

D. The execution plan is saved in DBA_HIST_SQL_PLAN without executing the query.

E. The execution plan generated can be viewed using the DBMS_XPLAIN.DISPLAY function.

F. The execution plan generated can be fetched from the library cache by using the DBMS_XPLAIN.DISPLAY function.

Correct Answer: ACE

*

(A, not D): The explain plan process stores data in the PLAN_TABLE.

*

EXPLAIN PLAN

The EXPLAIN PLAN method doesn\\'t require the query to be run (A), greatly reducing the time it takes to get an execution plan for long-running queries compared to AUTOTRACE.

E: Use the DBMS_XPLAN.DISPLAY function to display the execution plan.

* The DBMS_XPLAN package provides an easy way to display the output of the EXPLAIN PLAN command in several, predefined formats. You can also use the DBMS_XPLAN package to display the plan of a statement stored in the Automatic Workload Repository (AWR) or stored in a SQL tuning set. It further provides a way to display the SQL execution plan and SQL execution runtime statistics for cached SQL cursors based on the information stored in the V$SQL_PLAN and V$SQL_PLAN_STATISTICS_ALL fixed views.

Note:

*

First the query must be explained.

SQL> EXPLAIN PLAN FOR

2 SELECT *

```
3 FROM emp e, dept d

4 WHERE e.deptno = d.deptno

5 AND e.ename = \\'SMITH\\';

Explained.

SQL>
```

Then the execution plan displayed. (not B)

```
SQL> @$ORACLE_HOME/rdbms/admin/utlxpls.sql

Plan Table

-------------------------------------------------------------------------------- | Operation | Name | Rows | Bytes| Cost | Pstart| Pstop |

-------------------------------------------------------------------------------- | SELECT STATEMENT | | | | | | |

| NESTED LOOPS | | | | | | |

| TABLE ACCESS FULL |EMP | | | | | |

| TABLE ACCESS BY INDEX RO|DEPT | | | | | |

| INDEX UNIQUE SCAN |PK_DEPT | | | | | |

8 rows selected.

SQL>
```

For parallel queries use the "utlxplp.sql" script instead of "utlxpls.sql".

QUESTION 2

You identified some DSS queries that perform expensive join and aggregation operations.

The queries access historical data from noncurrent partition of the fact tables.

What three actions could you perform to improve the response time of the queries without modifying the SQL statements?

A. Set the QUERY_REWRITE_ENABLED to TRUE at the session level.

B. Create an STS for the statements, run SQL Tuning Advisor for the STS, and implement any generated recommendations for materialized views.

C. Set QUERY_REWRITE_ENABLED to TRUE at the instance level.

D. Create an STS for the statements, run SQL Access Advisor for the STS, and implement any generated recommendations for materialized views.

E. Set QUERY_REWRITE_INTEGRITY to ENFORCED at the instance level.

Correct Answer: BCD

A: * QUERY_REWRITE_ENABLED allows you to enable or disable query rewriting globally for the database. Values:

false

Oracle does not use rewrite.

true

Oracle costs the query with rewrite and without rewrite and chooses the method with the lower cost.

force

Oracle always uses rewrite and does not evaluate the cost before doing so. Use force when you know that the query will always benefit from rewrite and when

reduction in compile time is important.

To take advantage of query rewrite for a particular materialized view, you must enable query rewrite for that materialized view, and you must enable cost-based

optimization.

C: You can use SQL Tuning Advisor to tune one or more SQL statements

D: Using the SQL Access Advisor Wizard or API, you can do the following:

Recommend materialized views and indexes based on collected or hypothetical workload information.

Manage workloads.

Mark, update, and remove recommendations.

Note:

*

STS SQL tuning set.

*

A SQL Tuning Set is a database object that includes one or more SQL statements and their execution statistics and execution context. You can use the set as an input source for various advisors, such as SQL Tuning Advisor, SQL Access Advisor, and SQL Performance Analyzer.

Incorrect:

E: QUERY_REWRITE_INTEGRITY determines the degree to which Oracle must enforce query rewriting. At the safest level, Oracle does not use query rewrite transformations that rely on unenforced relationships.

Values:

enforced

Oracle enforces and guarantees consistency and integrity.

trusted

Oracle allows rewrites using relationships that have been declared, but that are not enforced by Oracle.

stale_tolerated

Oracle allows rewrites using unenforced relationships. Materialized views are eligible for rewrite even if they are known to be inconsistent with the underlying detail

data

QUESTION 3

You are administering a database supporting a DDS workload in which some tables are updated frequently but not queried often. You have SQL plan baseline for these tables and you do not want the automatic maintenance task to gather statistics for these tables regularly.

Which task would you perform to achieve this?

A. Set the INCREMENTAL statistic preference FALSE for these tables.

B. Set the STALE_PERCENT static preference to a higher value for these tables.

C. Set the GRANULARITY statistic preference to AUTO for these tables.

D. Set the PUBLISH statistic preference to TRUE for these tables.

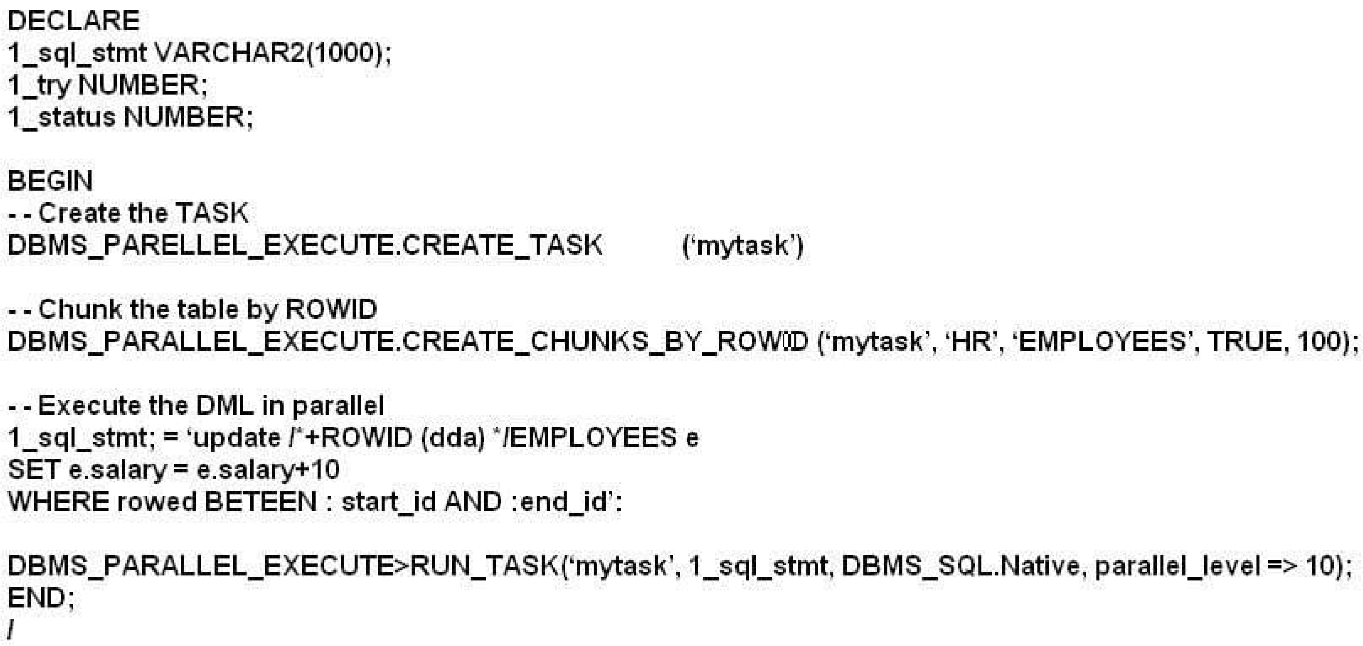
Correct Answer: B

With the DBMS_STATS package you can view and modify optimizer statistics gathered for database objects.

STALE_PERCENT - This value determines the percentage of rows in a table that have to change before the statistics on that table are deemed stale and should be regathered. The default value is 10%. Reference: Oracle Database PL/SQL Packages and Types Reference

QUESTION 4

Examine the following anonymous PL/SQL code block of code:



```
DECLARE
1_sql_stmt VARCHAR2(1000);
1_try NUMBER;
1_status NUMBER;

BEGIN
- - Create the TASK
DBMS_PARELLEL_EXECUTE.CREATE_TASK        ('mytask')

- - Chunk the table by ROWID
DBMS_PARALLEL_EXECUTE.CREATE_CHUNKS_BY_ROWID ('mytask', 'HR', 'EMPLOYEES', TRUE, 100);

- - Execute the DML in parallel
1_sql_stmt; = 'update /*+ROWID (dda) */EMPLOYEES e
SET e.salary = e.salary+10
WHERE rowed BETEEN : start_id AND :end_id':

DBMS_PARALLEL_EXECUTE>RUN_TASK('mytask', 1_sql_stmt, DBMS_SQL.Native, parallel_level => 10);
END;
/
```

Which two are true concerning the use of this code?

A. The user executing the anonymous PL/SQL code must have the CREATE JOB system privilege.

B. ALTER SESSION ENABLE PARALLEL DML must be executed in the session prior to executing the anonymous PL/SQL code.

C. All chunks are committed together once all tasks updating all chunks are finished.

D. The user executing the anonymous PL/SQL code requires execute privilege on the DBMS_JOB package.

E. The user executing the anonymous PL/SQL code requires privilege on the DBMS_SCHEDULER package.

F. Each chunk will be committed independently as soon as the task updating that chunk is finished.

Correct Answer: AE

A (not D, not E):

To use DBMS_PARALLEL_EXECUTE to run tasks in parallel, your schema will need the CREATE JOB system privilege.

E (not C): DBMS_PARALLEL_EXECUTE now provides the ability to break up a large table according to a variety of criteria, from ROWID ranges to key values and

user-defined methods. You can then run a SQL statement or a PL/SQL block against these different "chunks" of the table in parallel, using the database scheduler

to manage the processes running in the background. Error logging, automatic retries, and commits are integrated into the processing of these chunks.

Note:

*

The DBMS_PARALLEL_EXECUTE package allows a workload associated with a base table to be broken down into smaller chunks which can be run in parallel. This process involves several distinct stages. 1.Create a task 2.Split the workload into chunks CREATE_CHUNKS_BY_ROWID CREATE_CHUNKS_BY_NUMBER_COL CREATE_CHUNKS_BY_SQL 3.Run the task RUN_TASK User-defined framework Task control 4.Check the task status 5.Drop the task

*

The workload is associated with a base table, which can be split into subsets or chunks of rows. There are three methods of splitting the workload into chunks.

CREATE_CHUNKS_BY_ROWID CREATE_CHUNKS_BY_NUMBER_COL CREATE_CHUNKS_BY_SQL The chunks associated with a task can be dropped using the DROP_CHUNKS procedure.

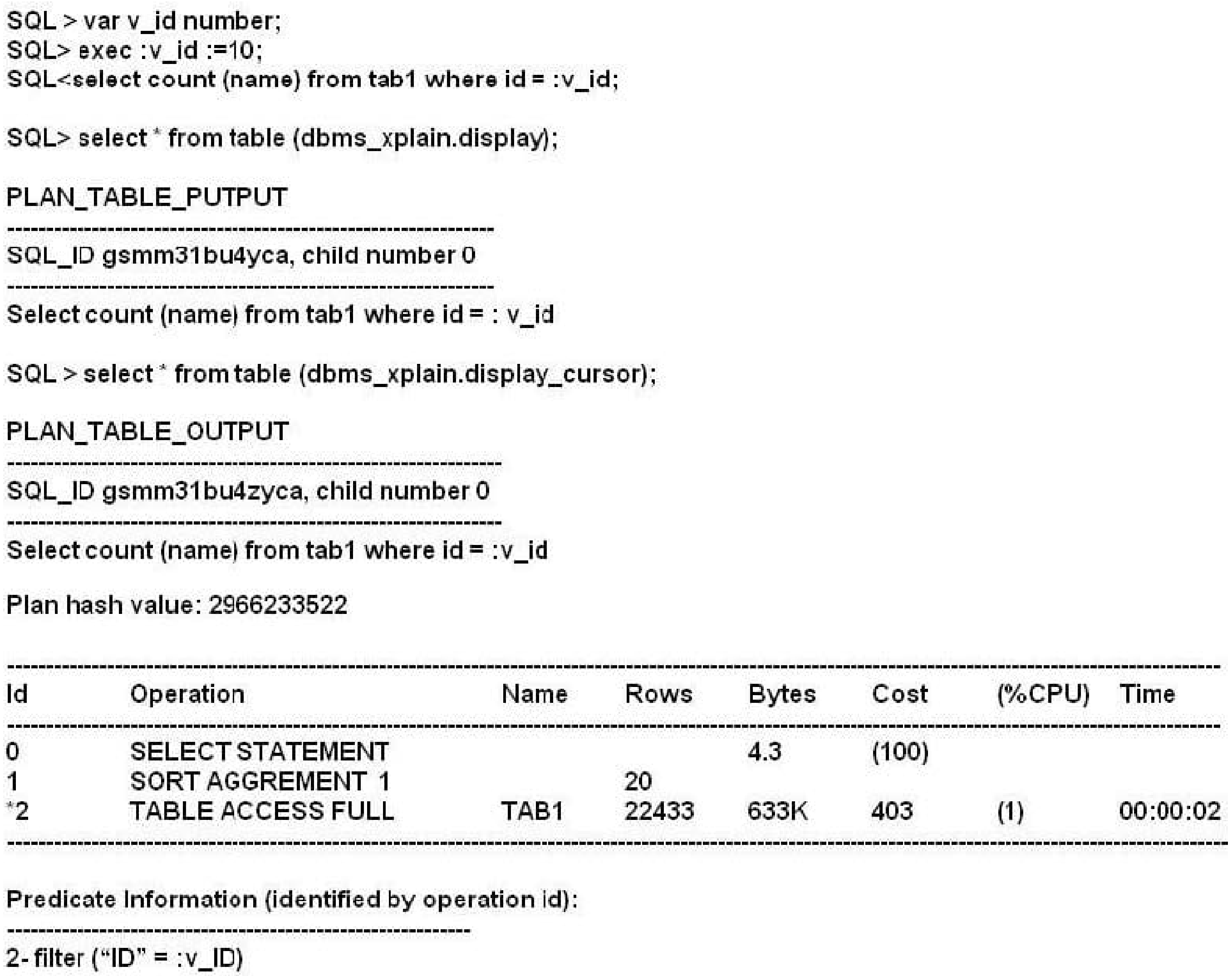
*

CREATE_CHUNKS_BY_ROWID The CREATE_CHUNKS_BY_ROWID procedure splits the data by rowid into chunks specified by the CHUNK_SIZE parameter. If the BY_ROW parameter is set to TRUE, the CHUNK_SIZE refers to the number of rows, otherwise it refers to the number of blocks.

Reference: TECHNOLOGY: PL/SQL Practices, On Working in Parallel

QUESTION 5

Exhibit



```
SQL > var v_id number;
SQL> exec :v_id :=10;
SQL<select count (name) from tab1 where id = :v_id;

SQL> select * from table (dbms_xplain.display);

PLAN_TABLE_PUTPUT
-------------------------------------------------------------
SQL_ID gsmm31bu4yca, child number 0
-------------------------------------------------------------
Select count (name) from tab1 where id = : v_id

SQL > select * from table (dbms_xplain.display_cursor);

PLAN_TABLE_OUTPUT
--------------------------------------------------------------
SQL_ID gsmm31bu4zyca, child number 0
--------------------------------------------------------------
Select count (name) from tab1 where id = :v_id

Plan hash value: 2966233522

--------------------------------------------------------------------------------------------------------------------------
Id        Operation                     Name      Rows      Bytes     Cost      (%CPU)    Time
--------------------------------------------------------------------------------------------------------------------------
0         SELECT STATEMENT                                  4.3       (100)
1         SORT AGGREMENT  1                       20
*2        TABLE ACCESS FULL             TAB1      22433     633K      403       (1)       00:00:02
--------------------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
----------------------------------------------------------
2- filter ("ID" = :v_ID)
```

A table has three distinct values in its ID column. In the ID column, values 10 and 20 have more than 20000 rows each and value 30 has only five rows. The

statistics for the schema have been updated recently.

The CURSOR_SHARING parameter is set to EXACT.

The query was executed recently and the cursor for the query is bind aware. Examine the exhibits to view the commands executed.

You plan to execute the same query with a value of 30 for the bind variable V_ID.

Which statement is true in this scenario?

A. The same execution plan will always be used irrespective of the value in the bind variable.

B. A new execution plan will be generated depending on the access pattern and the bind value.

C. Adaptive cursor sharing will ensure that a new cursor is generated for each distinct value in the bind variable.

D. Adaptive cursor sharing will happen only if you use the literal values instead of bind variables in the query.

Correct Answer: C

Note:

*

CURSOR_SHARING determines what kind of SQL statements can share the same cursors.

*

Setting CURSOR_SHARING to EXACT allows SQL statements to share the SQL area only when their texts match exactly. This is the default behavior. Using

this setting, similar statements cannot shared; only textually exact statements can be shared.

*

Values:

*

FORCE

Forces statements that may differ in some literals, but are otherwise identical, to share a cursor, unless the literals affect the meaning of the statement.

*

SIMILAR

Causes statements that may differ in some literals, but are otherwise identical, to share a cursor, unless the literals affect either the meaning of the statement or

the degree to which the plan is optimized.

*

EXACT

Only allows statements with identical text to share the same cursor.