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.passapply.com/1z0-117.html

100% Passing Guarantee
100% Money Back Assurance

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

Instant Download After Purchase
100% Money Back Guarantee
365 Days Free Update
800,000+ Satisfied Customers

QUESTION 1

You need to migrate database from oracle Database 10g to 11g. You want the SQL workload to start the 10g plans in the 11g database instance and evolve better plans.

Examine the following steps:

1.

Capture the pre-Oracle Database 11g plans in a SQL Tuning Set (STS)

2.

Export the STS from the 10g system.

3.

Import the STS into Oracle Database 11g.

4.

Set the OPTIMIZER_FEATURES_ENABLE parameter to 10.2.0.

5.

Run SQL Performance Analyzer for the STS.

6.

Set the OPTIMIZER_FEATURES_ENABLE parameter to 11.2.0.

7.

Rerun the SQL Performance Analyzer for the STS.

8.

Set OPTIMIZER_CAPTURE_SQL_PLAN_BASELINE to TRUE.

9.

Use DBMS_SPM.EVOLVE_SQL_BASELINE function to evolve the plans.

10.

Set the OPTIMIZER_USE_SQL_PLAN_BASELINE to TRUE.

Identify the required steps in the correct order.

A. 1, 2, 3, 4, 5, 6, 7,

B. 4, 8, 10

C. 1, 2, 3, 4, 8, 10

D. 1, 2, 3, 6, 9, 5

E. 1, 2, 3, 5, 9, 10

Correct Answer: C

Step 1: (1)

Step 2: (2)

Step 3: (3)

Step 4: (4)

By setting the parameter OPTIMIZER_FEATURES_ENABLE to the 10g version used before the upgrade, you should be able to revert back to the same execution

plans you had prior to the upgrade.

Step 5: (8)

OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES

In Oracle Database 11g a new feature called SQL Plan Management (SPM) has been introduced to guarantees any plan changes that do occur lead to better

performance. When OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES is set to TRUE (default FALSE) Oracle will

automatically capture a SQL plan baseline for every repeatable SQL statement on the system. The execution plan found at parse time will be added to the SQL

plan baseline as an accepted plan.

Step 6: (10)

OPTIMIZER_USE_SQL_PLAN_BASELINES enables or disables the use of SQL plan baselines stored in SQL Management Base. When enabled, the optimizer

looks for a SQL plan baseline for the SQL statement being compiled. If one is found in SQL Management Base, then the optimizer will cost each of the baseline

plans and pick one with the lowest cost.

QUESTION 2

Which two types of column filtering may benefit from partition pruning?

A. Equally operates on range-partitioned tables.

B. In-list operators on system-partitioned tables

C. Equality operators on system-partitioned tables

D. Operators on range-partitioned tables

E. Greater than operators on hash-partitioned tables

Correct Answer: AD

The query optimizer can perform pruning whenever a WHERE condition can be reduced to either one of the following two cases:

partition_column = constant

partition_column IN (constant1, constant2, ..., constantN)

In the first case, the optimizer simply evaluates the partitioning expression for the value given, determines which partition contains that value, and scans only this partition. In many cases, the equal sign can be replaced with another arithmetic comparison, including , =, and . Some queries using BETWEEN in the WHERE clause can also take advantage of partition pruning.

Note:

*

The core concept behind partition pruning is relatively simple, and can be described as "Do not scan partitions where there can be no matching values".

When the optimizer can make use of partition pruning in performing a query, execution of the query can be an order of magnitude faster than the same query against a nonpartitioned table containing the same column definitions and data.

*

Example:

Suppose that you have a partitioned table t1 defined by this statement:

CREATE TABLE t1 (

fname VARCHAR(50) NOT NULL,

lname VARCHAR(50) NOT NULL,

region_code TINYINT UNSIGNED NOT NULL,

dob DATE NOT NULL

)

PARTITION BY RANGE(region_code) (

PARTITION p0 VALUES LESS THAN (64),

PARTITION p1 VALUES LESS THAN (128),

PARTITION p2 VALUES LESS THAN (192),

PARTITION p3 VALUES LESS THAN MAXVALUE

);

Consider the case where you wish to obtain results from a query such as this one:

SELECT fname, lname, region_code, dob

FROM t1

WHERE region_code > 125 AND region_code

p0 or

It is easy to see that none of the rows which ought to be returned will be in either of the partitions p3; that is, we need to search only in partitions p1 and p2 to find

matching rows. By doing so, it is possible to expend much less time and effort in finding matching rows than would be required to scan all partitions "cutting away"

of unneeded partitions is known as pruning.

in the table. This

QUESTION 3

A new application module is deployed on middle tier and is connecting to your database. You want to monitor the performance of the SQL statements generated from the application.

To accomplish this, identify the required steps in the correct order from the steps given below:

1.

Use DBNMS_APPLICATION_INFO to set the name of the module

2.

Use DBMS_MONITOR.SERV_MOD_ACT_STAT_ENABLE to enable statistics gathering for the module.

3.

Use DBMS_MONITOR.SERV_MOD_ACT_TRACE_ENABLE to enable tracing for the service

4.

Use the trcsess utility to consolidate the trace files generated.

5.

Use the tkprof utility to convert the trace files into formatted output.

A. 1, 2, 3, 4, 5

B. 2, 3, 1, 4, 5

C. 3, 1, 2, 4, 5

D. 1, 2, 4, 5

E. 1, 3, 4, 5

F. 2, 1, 4, 5

Correct Answer: A

Note:

*

Before tracing can be enabled, the environment must first be configured to enable gathering of statistics.

*

(gather statistics): DBMS_MONITOR.SERV_MOD_ACT_STAT_ENABLE

Enables statistic gathering for a given combination of Service Name, MODULE and ACTION

*

DBMS_MONITOR.SERV_MOD_ACT_TRACE_ENABLE

Enables SQL tracing for a given combination of Service Name, MODULE and ACTION globally unless an instance_name is specified.

dbms_monitor.serv_mod_act_trace_enable(

service_name IN VARCHAR2,

module_name IN VARCHAR2 DEFAULT ANY_MODULE,

action_name IN VARCHAR2 DEFAULT ANY_ACTION,

waits IN BOOLEAN DEFAULT TRUE,

binds IN BOOLEAN DEFAULT FALSE,

instance_name IN VARCHAR2 DEFAULT NULL,

plan_stat IN VARCHAR2 DEFAULT NULL);

SELECT instance_name

FROM gv$instance;

exec dbms_monitor.serv_mod_act_trace_enable(\\'TESTSERV\\', dbms_monitor.all_modules, dbms_monitor.all_actions, TRUE, TRUE, \\'orabase\\');

exec dbms_monitor.serv_mod_act_trace_disable(\\'TESTSERV\\', dbms_monitor.all_modules, dbms_monitor.all_actions, \\'orabase\\');

*

When solving tuning problems, session traces are very useful and offer vital information. Traces are simple and straightforward for dedicated server sessions,

but for shared server sessions, many processes are involved. The trace pertaining to the user session is scattered across different trace files belonging to different

processes. This makes it difficult to get a complete picture of the life cycle of a session.

Now there is a new tool, a command line utility called trcsess to help read the trace files. The trcsess command-line utility consolidates trace information from

selected trace files, based on specified criteria. The criteria include session id, client id, service name, action name and module name.

*

Once the trace files have been consolidated (with trcsess), tkprof can be run against the consolidated trace file for reporting purposes.

QUESTION 4

An application supplied by a new vendor is being deployed and the SQL statements have plan baselines provided by the supplier. The plans have been loaded from a SQL tuning set. You require the optimizer to use these baselines, but allow better plans to used, should any be created.

Which two tasks would you perform to achieve this?

A. Set the OPTIMIZER_USE_SQL_PLAN_BASELINES initialization parameter to TRUE.

B. Set the OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES initialization parameter to TRUE.

C. Use the DBMS_SPM.ALTER_SQL_PLAN_BASELINE function to fix the plans.

D. Use the DBMS_SPM.EVOLVE_SQL_PLAN_BASELINE function to fix the new plans.

E. Use the DBMS_SPM.ALTER_SQL_BASELINE function to accept new plans.

Correct Answer: AD

A: OPTIMIZER_USE_SQL_PLAN_BASELINES enables or disables the use of SQL plan baselines stored in SQL Management Base. When enabled, the optimizer looks for a SQL plan baseline for the SQL statement being compiled. If one is found in SQL Management B ase, then the optimizer will cost each of the baseline plans and pick one with the lowest cost.

D: EVOLVE_SQL_PLAN_BASELINE Function

This function evolves SQL plan baselines associated with one or more SQL statements. A SQL plan baseline is evolved when one or more of its non-accepted plans is changed to an accepted plan or plans. If interrogated by the user (parameter verify = \\'YES\\'), the execution performance of each non-accepted plan is compared against the performance of a plan chosen from the associated SQL plan baseline. If the non-accepted plan performance is found to be better than SQL plan baseline performance, the non-accepted plan is changed to an accepted plan provided such action is permitted by the user (parameter commit = \\'YES\\').

Incorrect:

B: OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES enables or disables the automatic recognition of repeatable SQL statements, as well as the generation of SQL plan baselines for such statements.

C: ALTER_SQL_PLAN_BASELINE Function

This function changes an attribute of a single plan or all plans associated with a SQL statement using the attribute

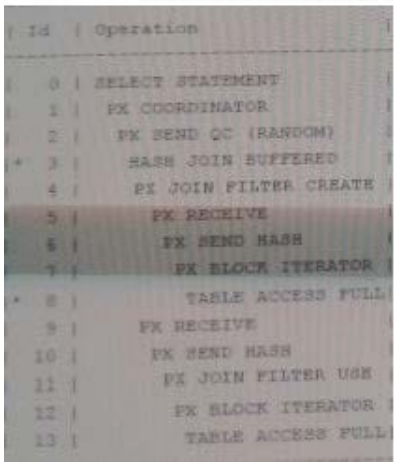
name/value format.

QUESTION 5

Examine the Exhibit.

Id	Operation	Name	TQ	IN-OUT	PQ Distrib
0	SELECT STATEMENT				
1	PX COORDINATOR				
2	PX SEND QC (RANDOM)	:TQ10002	Q1, 02	P->S	QC (RAND)
*3	HASH JOIN BUFFERED		Q1, 02	PCWP	
4	PX JOIN FILTER CREATE	:BF0000	Q1, 02	PCWP	
5	PX RECEIVE		Q1, 02	PCWP	
6	PX SEND HASH	:TQ10000	Q1, 00	P->P	HASH
7	PX BLOCK ITERATOR		Q1, 00	PCWP	
*8	TABLE ACCESS FULL	T1	Q1, 00	PCWP	
9	PX RECEIVE		Q1, 02	PCWP	
10	PX SEND HASH	:TQ10001	Q1, 01	P->P	HASH
11	PX JOIN FILTER USE	:BF0000	Q1, 01	PCWP	
12	PX BLOCK ITERATOR		Q1, 01	PCWC	
13	TABLE ACCESS FULL	T2	Q1, 01	PCWP	

3 – access ("T1". "ID" = "T2". "ID")
8 – filter ("T1". "MOD" = 42)



Which two statements are true about the bloom filter in the execution plan?

A. The bloom filter prevents all rows from table T1 that do not join T2 from being needlessly distributed.

B. The bloom filter prevents all rows from table T2 that do not join table T1 from being needlessly distributed.

C. The bloom filter prevents some rows from table T2 that do not join table T1 from being needlessly distributed.

D. The bloom filter is created in parallel by the set of parallel execution processes that scanned table T2.

E. The bloom filter is created in parallel by the set of parallel execution processes that later perform join.

F. The bloom filter is created in parallel by the set of parallel execution processes that scanned table T1.

Correct Answer: BF

*

PX JOIN FILTER CREATE The bloom filter is created in line 4.

*

PX JOIN FILTER USE The bloom filter is used in line 11.

Note:

*

You can identify a bloom pruning in a plan when you see :BF0000 in the Pstart and Pstop columns of the execution plan and PART JOIN FILTER CREATE in the operations column.

*

A Bloom filter is a probabilistic algorithm for doing existence tests in less memory than a full list of keys would require. In other words, a Bloom filter is a method for representing a set of n elements (also called keys) to support membership queries.

*

The Oracle database makes use of Bloom filters in the following 4 situations:

-To reduce data communication between slave processes in parallel joins: mostly in RAC

-

To implement join-filter pruning: in partition pruning, the optimizer analyzes FROM and WHERE clauses in SQL statements to eliminate unneeded partitions when building the partition access list

-

To support result caches: when you run a query, Oracle will first see if the results of that query have already been computed and cached by some session or

user, and if so, it will retrieve the answer from the server result cache instead of gathering all of the database blocks

-

To filter members in different cells in Exadata: Exadata performs joins between large tables and small lookup tables, a very common scenario for data warehouses with star schemas. This is implemented using Bloom filters as to determine

whether a row is a member of the desired result set.

1Z0-117 VCE Dumps 1Z0-117 Practice Test 1Z0-117 Study Guide