QUESTIONS & ANSWERS

Kill your exam at first Attempt

KILL EXAMS

Oracle

1Z0-117

Oracle Database 11g Release 2- SQL Tuning

OracleDatabase Data Warehousing Guide,Star Transformation with a Bitmap Index

QUESTION: 118
Which two statements are true about index full scans?

A. An index fast full scan multi block I/O to read the index structure in its entirely.
B. Index nodes are not retrieved in the index order, and there fore the nodes are not in sequence.
C. An index fast full scan reads the index block by block.
D. An index fast full scan reads the whole index from the lowest value to the higher value.

Answer: A, B

Explanation:
A:To speed table and index block access, Oracle uses the db_file_multiblock_read_count parameter (which defaults to 8) to aid in getting full-table scan and full-index scan data blocks into the data buffer cache as fast as possible.
B:The index nodes are not retrieved in index order, the rows will not be sequenced.
Note:
there are some requirements for Oracle to invoke the fast full-index scan.

Reference:
index fast full scan tips

QUESTION: 119
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.

Answer: A, D

Explanation:
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: 120
You recently gathered statistics for a table by using the following commands:

```
SQL> exec SBMS.SET_TABLE_PREFS ('SH', 'CUSTOMERS', 'PUBLISH', 'TRUE');
SQL> exec DBMS_STATS.GATHER_TABLE_STATS ('SH', 'CUSTOMERS', NULL, 20, FALSE, 'FOR
ALLCOLUMNS', 4, 'DEFAULT, TRUE');
```

You noticed that the performance of queries has degraded after gathering statistics. You want to use the old statistics. The optimizer statistics retention period is default. What must you do to use the old statistics?

A. Use the flashback to bring back the statistics to the desired time.
B. Restore statistics from statistics history up to the desired time.
C. Delete all the statistics collected after the desired time.
D. Set OPTIMIZER_USE_PENDING_STATISTICS to TRUE.

Answer: B

Explanation:
Whenever statistics in dictionary are modified, old versions of statistics are saved automatically for future restoration. Statistics can be restored using RESTORE procedures of DBMS_STATS package. These procedures use a time stamp as an argument and restore statistics as of that time stamp. This is useful in case newly collected statistics leads to some sub- optimal execution plans and the administrator wants to revert to the previous set of statistics.

Reference:
Oracle Database Performance Tuning Guide,Restoring Previous Versions of Statistics

QUESTION: 121
View the exhibit and examine the plans in the SQL baseline for a given statement.

```
SELECT * FROM TABLE (
DBMS_XPLAIN.DISPLAY_SQL_BASELINE(
                    Sql_handle=> 'SYS_SQL_209d10fabbedc741',
                    Format => 'basic,));

SQL handle: SYS_SQL_209d10fabbdc741

SQL text: select cust_last_name, amount_sold from customers c,
          Sales s where c.cust_id = s.count and cust_year_of_birth = : yob

--------------------------------------------------------------------------------
Plan name: SYS_PLAN_BBEDC741a57bfc2
Enabled: YES        Fixed: NO Accepted: NO        Origin: AUTO_CAPTURE
--------------------------------------------------------------------------------
Plan hash value: 2776326082

-------------------------------------------------------------------------------
Id      Operation                           Name
-------------------------------------------------------------------------------
0       SELECT STATEMENT

1       HASH JOIN

2       TABLE ACCESS BY INDEX ROWID         CUSTOMERS

3       BITMAP CONVERSION

4       BITMAP INDEX SINGLE VALUE           CUSTOMER_YOB_BIX

5       PARTITION RANGE   ALL

6       TABLE ACCESS FULL                   SALES
-------------------------------------------------------------------------------

Plan Name: SYS_SQL_PLAN_BBDC741f554c408
Enabled: YES        Fixed: NO Accepted: YES       Origin: MANUAL_LOAD
--------------------------------------------------------------------------------
Plan hash value: 4115973128
----------------------------------------

Id      Operation                                   Name
-------------------------------------------------------------------------------
0       SELECT STATEMENT

1       NESTED LOOPS

2       NESTED LOOPA

3       TABLE ACCESS BY INDEX ROWID                 CUSTOMERS

4       BITMAP CONVERSION TO WODIDS

5       BITMAP CONVERSION TO ROWIDS                 CUSTOMERS_YOB_BIX

6       PARTITION RANGE

7       BITMAP CONVERSION TO ROWIDS

8       BITMAP INDEX SINGLE VALUE                   SALES_CUST_BIX

9       TABLE ACCESS BY LOCAL INDEX ROWIDS                  SALES
-------------------------------------------------------------------------------
```

Which interpretation is correct?

A. A new plan cannot be evolved because SYS_SQL_bbedc41f554c408 is accepted.
B. Plan SYS_SQL_PLAN_bbdc741f554c408 will always be used by the optimizer for the query.
C. A new plan must be evolved using the DBMS_SPM.EVOLVE_SQL_PLAN_BASELINE function before it can be used.
D. Plan SYS_SQL_bbedc741a57b5fc2 can be used by the optimizer if the cost of the query is less than plan SYS_SQL_PLAN_bbedc741f554c408.
E. Plan SYS_SQL_PLAN_bbedc741f554c408 will not be used until it is fixed by using the DBMS_SPM.EVOLVE_SQL_PLAN_BASELINE function.

Answer: C

Explanation:
Note:
*Evolving a SQL plan baseline is the process by which the optimizer determines if non-accepted plans in the baseline should be accepted. As mentioned previously, manually loaded plans are automatically marked as accepted, so manual loading forces the evolving process. When plans are loaded automatically, the baselines are evolved using the EVOLVE_SQL_PLAN_BASELINE function, which returns a CLOB reporting its results.
SET LONG 10000
SELECT DBMS_SPM.evolve_sql_plan_baseline(sql_handle => 'SYS_SQL_7b76323ad90440b9') FROM dual;
*Manual plan loading can be used in conjunction with, or as an alternative to automatic plan capture. The load operations are performed using the DBMS_SPM package, which allows SQL plan baselines to be loaded from SQL tuning sets or from specific SQL statements in the cursor cache. Manually loaded statements are flagged as accepted by default. If a SQL plan baseline is present for a SQL statement, the plan is added to the baseline, otherwise a new baseline is created.
*fixed (YES/NO) : If YES, the SQL plan baseline will not evolve over time. Fixed plans are used in preference to non-fixed plans.

QUESTION: 122
You want to run SQL Tuning Advisor statements that are not captured by ADDM, AWR, and are not in the library cache. What is the prerequisite?

A. Enable SQL plan management
B. Create a SQL plan baseline for each query
C. Create a SQL Tuning Set (STS) containing the SQL statements
D. Gather statistics for objects used in the application

Answer: C

Explanation:
You can use an STS as input to SQL Tuning Advisor, which performs automatic tuning of the SQL statements based on other user-specified input parameters.
Note:
A SQL tuning set (STS) is a database object that includes one or more SQL statements along with their execution statistics and execution context, and could include a user priority ranking. You can load SQL statements into a SQL tuning set from different SQL sources, such as AWR, the shared SQL area, or customized SQL provided by the user. An STS includes:

A set of SQL statements
Associated execution context, such as user schema, application module name and action, list of bind values, and the cursor compilation environment
Associated basic execution statistics, such as elapsed time, CPU time, buffer gets, disk reads, rows processed, cursor fetches, the number of executions, the number of complete executions, optimizer cost, and the command type
Associated execution plans and row source statistics for each SQL statement (optional).

Reference:
OracleDatabase Performance Tuning Guide,Managing SQL Tuning Sets

QUESTION: 123
While tuning a SQL statement, the SQL Tuning Advisor finds an existing SQL profile for a statement that has stale statistics. Automatic optimizer statistics is enabled for the database. What does the optimizer do in this situation?

A. Updates the existing SQL profiles for which the statistics are stale.
B. Makes the statistics information available to GATHER_DATABASE_STATS_JOB_PROC
C. Starts the statistics collection process by running GATHER_STATS_JOB
D. Writes a warning message in the alert log file

Answer: B

Explanation:
Automatic optimizer statistics collection calls
the DBMS_STATS.GATHER_DATABASE_STATS_JOB_PROC procedure.
This internal procedure operates similarly to the DBMS_STATS.GATHER_DATABASE_STATS procedure using the GATHER AUTO option. The main difference is
that GATHER_DATABASE_STATS_JOB_PROCprioritizes database objects that require statistics, so that objects that most need updated statistics are processed first, before the maintenance
window closes.
Note:
*The optimizer relies on object statistics to generate execution plans. If these statistics are stale or missing, then the optimizer does not have the necessary information it needs and can generate poor execution plans. The Automatic Tuning Optimizer checks each query object for missing or stale statistics, and produces two types of output:
/Recommendations to gather relevant statistics for objects with stale or no statistics

Because optimizer statistics are automatically collected and refreshed, this problem occurs only when automatic optimizer statistics collection is disabled. See "Managing Automatic Optimizer Statistics Collection".
/Auxiliary statistics for objects with no statistics, and statistic adjustment factor for objects with stale statistics
The database stores this auxiliary information in an object called a SQL profile.
*Oracle recommends that you enable automatic optimizer statistics collection. In this case, the database automatically collects optimizer statistics for tables with absent or stale statistics. If fresh statistics are required for a table, then the database collects them both for the table and associated indexes.
Automatic collection eliminates many manual tasks associated with managing the optimizer. It also significantly reduces the risks of generating poor execution plans because of missing or stale statistics.
Automatic optimizer statistics collection calls the DBMS_STATS.GATHER_DATABASE_STATS_JOB_PROC procedure. This internal procedure operates similarly to the DBMS_STATS.GATHER_DATABASE_STATS procedure using the GATHER AUTO option. The main difference is that GATHER_DATABASE_STATS_JOB_PROC prioritizes database objects that require statistics,so that objects that most need updated statistics are processed first, before the maintenance window closes.

Reference:
OracleDatabase Performance Tuning Guide,Managing Automatic Optimizer Statistics Collection

QUESTION: 124
Refer to the Exhibit.

```
SQL> DESC stored

Name                          Null?            Type
----------------------------------------------------------------------------------------
STORE_ID                      NOT NULL         NUMBER (4)
STORE_NAME                                     VARCHAR2 (12)
STORE_ADDRESS                                  VARCHAR2(20)
START_DATE                                     DATE

SQL> DESC Sales

NAME                NULL?             TYPE
-------------------------------------------------------------------------
SALES_ID            NOT NULL          NUMBER(4)
ITEM_ID                               NUMBER(4)
UQANTITY                              NUMBER (10)
SALES_DATE                            DATE
STORE_ID                              NUMBER (4)
```

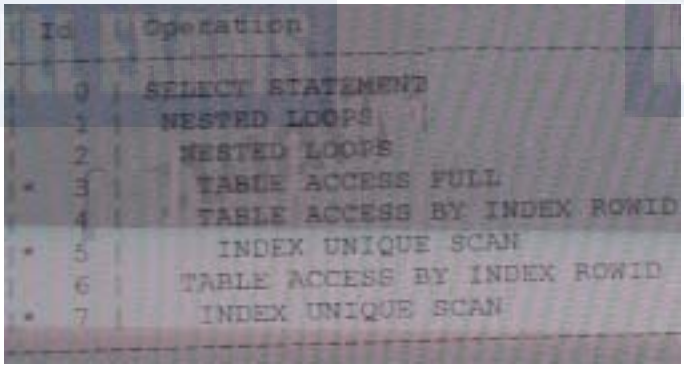
Execution plan:

Plan hash value: 29632623819

Id	Operation	Name	Rows	Bytes	Cost	(%CPU)
0	SELECT STATEMENT		3	189	10	(10)
1	NESTED LOOPS		3	189	10	(10)
2	NESTED LOOPS		3	141	7	(15)
*3	TABLE ACCESS FULL	EMPLOYEES	3	60	4	(25)
4	TABLE ACCESS BY INDEX ROWID	JOBS	19	513	2	(50)
*5	INDEX UNIQUE SCAN	JOB_ID_PK	1			
6	TABLE ACCESS BY INDEX ROWID	DEPARTMENTS	27	432	2	(50)
7	INDEX UNIQUE SCAN	DEPT_ID_PK	1			

PREDICATE Information (identified by operation id):

3 – filter ("E". "EMPLOYEE_ID"<103)
5 – access ("E". "JOB_ID" = "J_ID")
7 – access ("E". "DEPARTMENT_ID" = "D". "DEPARTMENT_ID")



What must be the correct order of steps that the optimizer executes based on the ID column the execution plan?

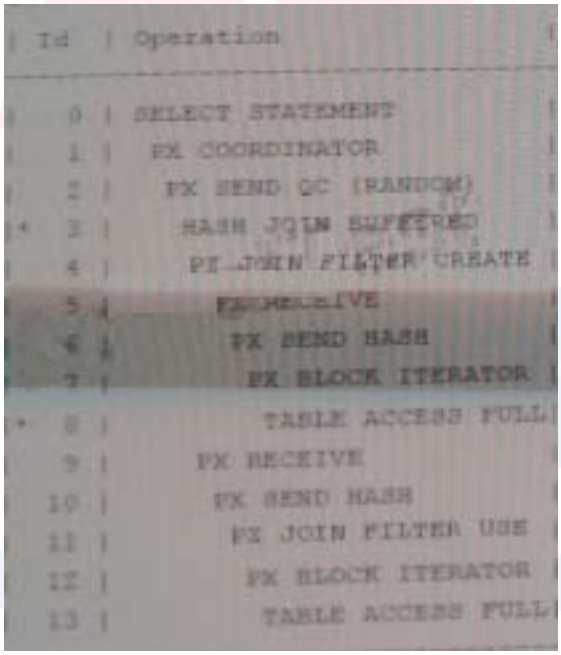
A. 3, 5, 4, 6, 7
B. 3, 5, 4, 7, 6
C. 3, 4, 5, 7, 6
D. 4, 5, 3, 7, 6

Answer: D

QUESTION: 125
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.

Answer: B, F

Explanation:
* 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 FILTERCREATE 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.