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

Oracle Database 11g Release 2: SQL Tuning Exam

QUESTION 1

Which three options are true about parallel queries when PARALLEL_DEGREE_POLICY is set to MANUAL and the session is using the default settings for parallel query, DDL, and DML?

A. A subquery in a parallel DML is parallelized only if it includes a parallel hint.

B. The number of parallel execution servers requested for a cursor is based on the greatest degree of parallelism associated with any object accessed by the cursor.

C. A SELECT statement can be executed in parallel only if no scalar subqueries are contained in the SELECT list.

D. In a CREATE TABLE . . . AS SELECT (CTAS) statement, SELECT is parallelized only if create TABLE is parallelized.

E. In an INSERT INTO . . . SELECT FROM statement, INSERT is parallelized if select is parallelized.

F. Single row inserts are never executed is parallel.

Correct Answer: CEF

* Decision to Parallelize

A SELECT statement can be parallelized only if the following conditions are satisfied:

/ The query includes a parallel hint specification (PARALLEL or PARALLEL_INDEX) or the schema objects referred to in the query have a PARALLEL declaration associated with them. / At least one of the tables specified in the query requires one of the following:

A full table scan

An index range scan spanning multiple partitions

/ (C) No scalar subqueries are in the SELECT list.

*

By default, the system only uses parallel execution when a parallel degree has been explicitly set on an object or if a parallel hint is specified in the SQL statement.

*

CREATE TABLE ... AS SELECT in Parallel

Parallel execution lets you parallelize the query and create operations of creating a table as a subquery from another table or set of tables. This can be extremely useful in the creation of summary or rollup tables.

Clustered tables cannot be created and populated in parallel.

* PARALLEL_DEGREE_POLICY specifies whether or not automatic degree of Parallelism, statement queuing, and in-memory parallel execution will be enabled.

MANUAL

Disables automatic degree of parallelism, statement queuing, and in-memory parallel execution. This reverts the behavior of parallel execution to what it was prior to Oracle Database 11g Release 2 (11.2). This is the default.

Incorrect:

A:

*

For parallel DML (INSERT, UPDATE, MERGE, and DELETE), the reference object that determines the DOP (degree of parallelism) is the table being modified by and insert, update, or delete operation. Parallel DML also adds some limits to the DOP to prevent deadlock. If the parallel DML statement includes a subquery, the subquery\\'s DOP is the same as the DML operation.

*

For parallel DDL, the reference object that determines the DOP is the table, index, or partition being created, rebuilt, split, or moved. If the parallel DDL statement includes a subquery, the subquery\\'s DOP is the same as the DDL operation.

D: The CREATE TABLE ... AS SELECT statement contains two parts: a CREATE part (DDL) and a SELECT part (query). Oracle Database can parallelize both parts of the statement. The query part of a CREATE TABLE ... AS SELECT statement can be parallelized only if the following conditions are satisfied:

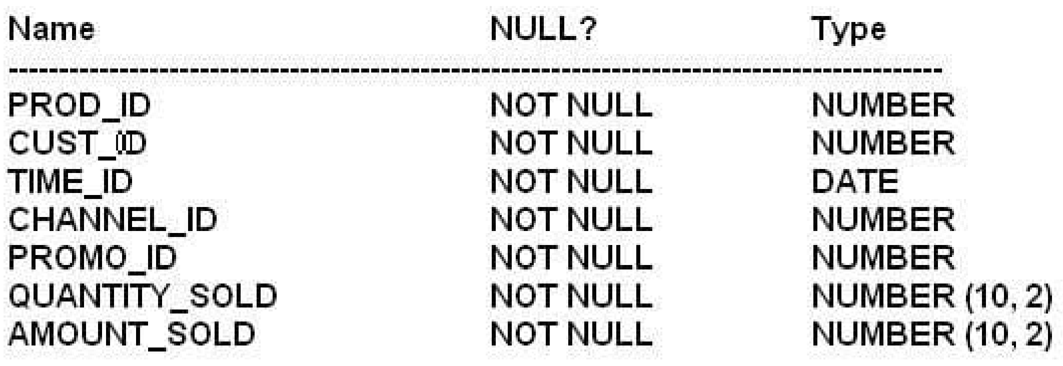
The query includes a parallel hint specification (PARALLEL or PARALLEL_INDEX) or the CREATE part of the statement has a PARALLEL clause specification or the schema objects referred to in the query have a PARALLEL declaration associated with them.

At least one of the tables specified in the query requires one of the following: a full table scan or an index range scan spanning multiple partitions.

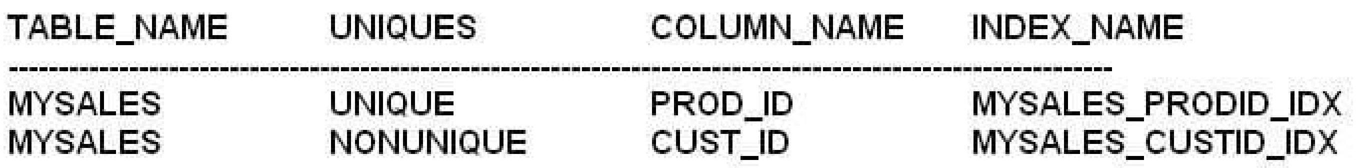
Reference: Oracle Database VLDB and Partitioning Guide, Using Parallel Execution

QUESTION 2

View the Exhibit1 and examine the structure and indexes for the MYSALES table.



Name	NULL?	Type
PROD_ID	NOT NULL	NUMBER
CUST_ID	NOT NULL	NUMBER
TIME_ID	NOT NULL	DATE
CHANNEL_ID	NOT NULL	NUMBER
PROMO_ID	NOT NULL	NUMBER
QUANTITY_SOLD	NOT NULL	NUMBER (10, 2)
AMOUNT_SOLD	NOT NULL	NUMBER (10, 2)



TABLE_NAME	UNIQUES	COLUMN_NAME	INDEX_NAME
MYSALES	UNIQUE	PROD_ID	MYSALES_PRODID_IDX
MYSALES	NONUNIQUE	CUST_ID	MYSALES_CUSTID_IDX

The application uses the MYSALES table to insert sales record. But this table is also extensively used for generating sales reports. The PROD_ID and CUST_ID columns are frequently used in the WHERE clause of the queries. These columns are frequently used in WHERE clause of the queries. These columns have few distinct values relative to the total number of rows in the table.

View exhibit 2 and examine one of the queries and its auto trace output. What should you do to improve the performance of the query?

A. Use the INDEX_COMBINE hint in the query.

B. Create composite index involving the CUST_ID and PROD_ID columns.

C. Gather histograms statistics for the CUST_ID and PROD_ID columns.

D. Gather index statistics for the MYSALES_PRODID_IDX and MYSALES_CUSTID_IDX indexes.

Correct Answer: D

Note:

*

Statistics quantify the data distribution and storage characteristics of tables, columns, indexes, and partitions.

*

INDEX_COMBINE Forces a bitmap index access path on tab. Primarily this hint just tells Oracle to use the bitmap indexes on table tab. Otherwise Oracle will choose the best combination of indexes it can think of based on the statistics. If it is ignoring a bitmap index that you think would be helpful, you may specify that index plus all of the others taht you want to be used. Note that this does not force the use of those indexes, Oracle will still make cost based choices.

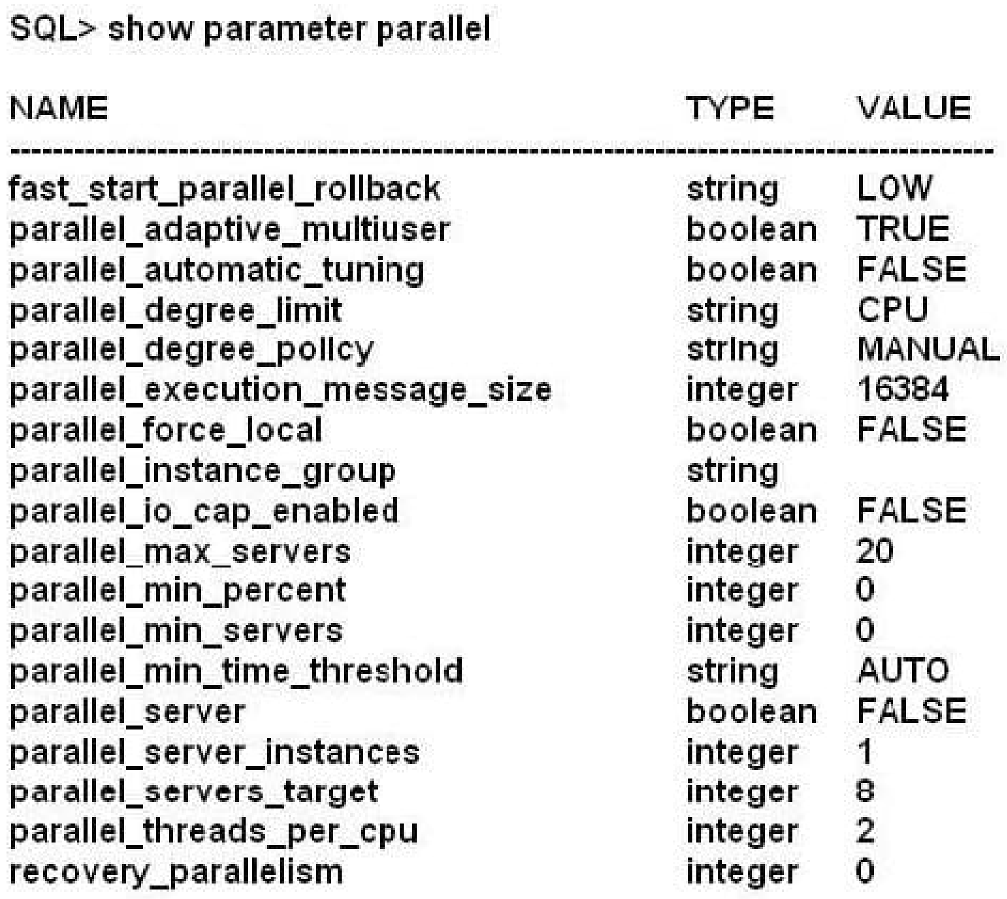
*

Histograms Opportunities Any column used in a where clause with skewed data Histograms are NOT just for indexed columns.

Adding a histogram to an un-indexed column that is used in

QUESTION 3

See the table below:



```
SQL> show parameter parallel

NAME                                 TYPE      VALUE
------------------------------------------------------------------------
fast_start_parallel_rollback         string    LOW
parallel_adaptive_multiuser          boolean   TRUE
parallel_automatic_tuning            boolean   FALSE
parallel_degree_limit                string    CPU
parallel_degree_policy               string    MANUAL
parallel_execution_message_size      integer   16384
parallel_force_local                 boolean   FALSE
parallel_instance_group              string
parallel_io_cap_enabled              boolean   FALSE
parallel_max_servers                 integer   20
parallel_min_percent                 integer   0
parallel_min_servers                 integer   0
parallel_min_time_threshold          string    AUTO
parallel_server                      boolean   FALSE
parallel_server_instances            integer   1
parallel_servers_target              integer   8
parallel_threads_per_cpu             integer   2
recovery_parallelism                 integer   0
```

All execution servers are currently available and the sessions use defaults for all parallel settings.

In which two cases will statements execute in parallel?

A. When parallel hints are used but only if estimated serial execution takes more than 10 seconds.

B. When parallelism is defined at the statement level.

C. When the degree of parallelism is explicitly defined in the data dictionary for tables and indexes accessed by a query.

D. Parallel DDL statements but only if estimated serial DDL execution time is greater than 10 seconds.

E. When the degree of parallelism is explicitly defined for tables and indexes but only if estimated serial execution takes more than 10 seconds.

Correct Answer: BC

Incorrect:

A, D, E: When PARALLEL_MIN_TIME_THRESHOLD is set to AUTO the PARALLEL_MIN_TIME_THRESHOLD is set to 30, not to 10. See note below.

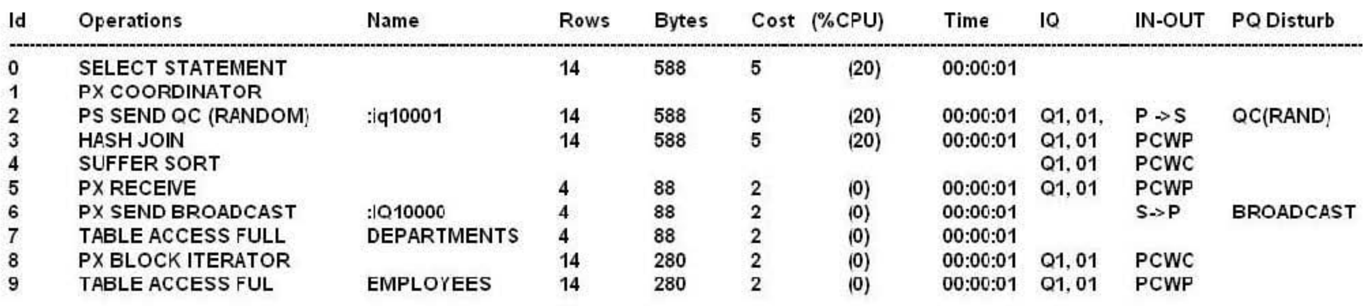
Note:

* parallel_min_time_threshold

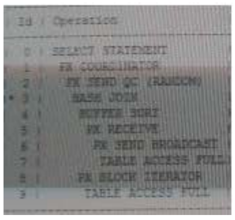
PARALLEL_MIN_TIME_THRESHOLD specifies the minimum execution time a statement should have before the statement is considered for automatic degree of parallelism. By default, this is set to 30 seconds. Automatic degree of parallelism is only enabled if PARALLEL_DEGREE_POLICY is set to AUTO or LIMITED.

QUESTION 4

Examine the exhibit.



Id	Operations	Name	Rows	Bytes	Cost (%CPU)	Time	IQ	IN-OUT	PQ Disturb
0	SELECT STATEMENT		14	588	5 (20)	00:00:01			
1	PX COORDINATOR								
2	PS SEND QC (RANDOM)	:iq10001	14	588	5 (20)	00:00:01	Q1, 01,	P ->S	QC(RAND)
3	HASH JOIN		14	588	5 (20)	00:00:01	Q1, 01	PCWP	
4	SUFFER SORT						Q1, 01	PCWC	
5	PX RECEIVE		4	88	2 (0)	00:00:01	Q1, 01	PCWP	
6	PX SEND BROADCAST	:IQ10000	4	88	2 (0)	00:00:01		S->P	BROADCAST
7	TABLE ACCESS FULL	DEPARTMENTS	4	88	2 (0)	00:00:01			
8	PX BLOCK ITERATOR		14	280	2 (0)	00:00:01	Q1, 01	PCWC	
9	TABLE ACCESS FUL	EMPLOYEES	14	280	2 (0)	00:00:01	Q1, 01	PCWP	



```
 Id | Operation
----------------------------
  0 | SELECT STATEMENT
  1 |  PX COORDINATOR
  2 |   PX SEND QC (RANDOM)
* 3 |    HASH JOIN
  4 |     BUFFER SORT
  5 |      PX RECEIVE
  6 |       PX SEND BROADCAST
  7 |        TABLE ACCESS FULL
  8 |     PX BLOCK ITERATOR
  9 |      TABLE ACCESS FULL
```

Which is true based on the information obtainable from the execution plan?

A. A full partition-wise join performed between the EMPLOYEES and DEPARTMENTS tables.

B. A full table scan on the DEPARTMENTS table performed serially by the query coordinator.

C. A full table scan on the DEPARTMENTS table is performed serially by a single parallel execution server process.

D. A partial partition-wise join performed between the EMPLOYEES and DEPARTMENTS tables.

E. A full table scan on the EMPLOYEES table is done in parallel.

Correct Answer: E

PX BLOCK ITERATOR This operation is typically the first step in a parallel pipeline. The BLOCK ITERATOR breaks up the table into chunks that are processed by

each of the parallel servers involved.

Incorrect:

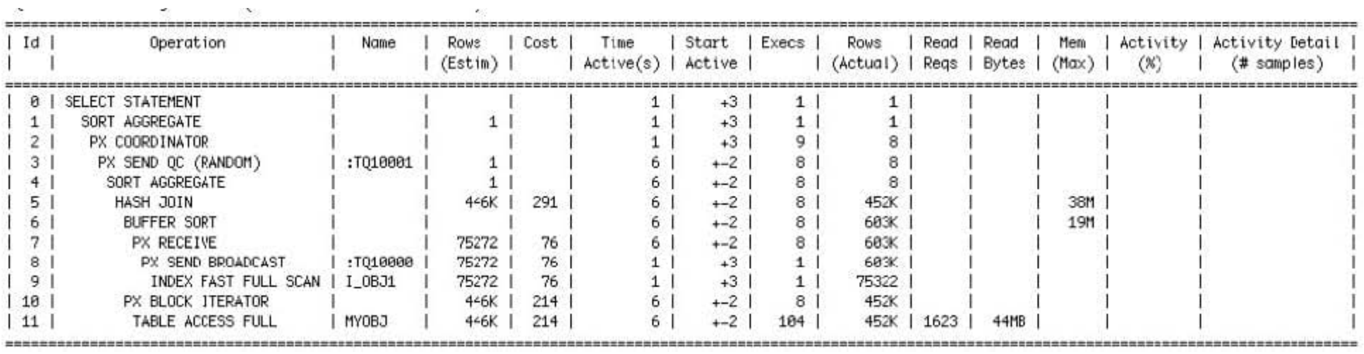
B, C: The scan on the Departsments table is done in parallel.

Note:

* As per exhibit: Line 7 is run first, followed by line 6.

*

Example with same structure of execution plan:



```
=======================================================================================================================================================================
| Id |          Operation           |   Name    |  Rows   | Cost |    Time    | Start  | Execs |   Rows   | Read | Read  |  Mem  | Activity | Activity Detail |
|    |                              |           | (Estim) |      | Active(s)  | Active |       | (Actual) | Reqs | Bytes | (Max) |   (%)    |  (# samples)    |
=======================================================================================================================================================================
|  0 | SELECT STATEMENT             |           |         |      |          1 |     +3 |     1 |        1 |      |       |       |          |                 |
|  1 |  SORT AGGREGATE              |           |       1 |      |          1 |     +3 |     1 |        1 |      |       |       |          |                 |
|  2 |   PX COORDINATOR             |           |         |      |          1 |     +3 |     9 |        8 |      |       |       |          |                 |
|  3 |    PX SEND QC (RANDOM)       | :TQ10001  |       1 |      |          6 |    +-2 |     8 |        8 |      |       |       |          |                 |
|  4 |     SORT AGGREGATE           |           |       1 |      |          6 |    +-2 |     8 |        8 |      |       |       |          |                 |
|  5 |      HASH JOIN               |           |    446K |  291 |          6 |    +-2 |     8 |     452K |      |       |   38M |          |                 |
|  6 |       BUFFER SORT            |           |         |      |          6 |    +-2 |     8 |     603K |      |       |   19M |          |                 |
|  7 |        PX RECEIVE            |           |   75272 |   76 |          6 |    +-2 |     8 |     603K |      |       |       |          |                 |
|  8 |         PX SEND BROADCAST    | :TQ10000  |   75272 |   76 |          1 |     +3 |     1 |     603K |      |       |       |          |                 |
|  9 |          INDEX FAST FULL SCAN | I_OBJ1   |   75272 |   76 |          1 |     +3 |     1 |    75322 |      |       |       |          |                 |
| 10 |       PX BLOCK ITERATOR      |           |    446K |  214 |          6 |    +-2 |     8 |     452K |      |       |       |          |                 |
| 11 |        TABLE ACCESS FULL     | MYOBJ     |    446K |  214 |          6 |    +-2 |   104 |     452K | 1623 |  44MB |       |          |                 |
=======================================================================================================================================================================
```

Here\\'s how to read the plan:

1.

The first thing done is at line 9 an index fast full scan on SYS.OBJ$.I_OBJ1 index. This is done in parallel, as indicated from the "PX SEND" line above.

2.

In line 8, we\\'re doing a "PX SEND BROADCAST" operation. When joining tables in parallel, Oracle can choose to either broadcast results (rows) from one operation to apply to the other table scan, or it can choose PX SEND HASH. In this case, our CBO determined that a BROADCOAST was appropriate because the results from the OBJ$ table were much lower than the MYOBJ table

3.

Line 7, the PX RECEIVE step, is basically the consumer of the broadcasted rows in step 8

4.

Line 6 is an in-memory BUFFER SORT of the rows returned from the index scan on OBJ$

5.

Lines 11 and 10, respectively, indicate the full scan and PX BOCK ITERATOR operation for the granules involved in the 8 PQ servers

6.

In line 5, Oracle is doing a hash join on the resulting rows from the parallel scans on MYOBJ and OBJ$

7.

Line 4 is a per-PQ server sort of data from the joind PQ servers

8.

Line 3 is the consumer QC that holds the result of the each of the PQ servers

9.

Line 2 is the PX Coordinator (QC) collecting, or consuming the rows of the joined data

10.

Line 1 is the final SORT AGGREGATE line that performs the grouping function

QUESTION 5

Which are the two prerequisites for enabling star transformation on queries?

A. The STAR_TRANSFORMATION_ENABLED parameter should be set to TRUE or TEMP_DISABLE.

B. A B-tree index should be built on each of the foreign key columns of the fact table(s),

C. A bitmap index should be built on each of the primary key columns of the fact table(s).

D. A bitmap index should be built on each of the foreign key columns of the fact table(s).

E. A bitmap index must exist on all the columns that are used in the filter predicates of the query.

Correct Answer: AE

A: Enabling the transformation

E: Star transformation is essentially about adding subquery predicates corresponding to the constraint dimensions. These subquery predicates are referred to as bitmap semi-join predicates. The transformation is performed when there are indexes on the fact join columns (s.timeid, s.custid...). By driving bitmap AND and OR operations (bitmaps can be from bitmap indexes or generated from regular B-Tree indexes) of the key values supplied by the subqueries, only the relevant rows from the fact table need to be retrieved. If the filters on the dimension tables filter out a lot of data, this can be much more efficient than a full table scan on the fact table. After the relevant rows have been retrieved from the fact table, they may need to be joined back to the dimension tables, using the original predicates. In some cases, the join back can be eliminated.

Star transformation is controlled by the star_transformation_enabled parameter. The parameter takes 3 values.

TRUE - The Oracle optimizer performs transformation by identifying fact and constraint dimension tables automatically. This is done in a cost-based manner, i.e.

the transformation is performed only if the cost of the transformed plan is lower than the non-transformed plan. Also the optimizer will attempt temporary table

transformation automatically whenever materialization improves performance.

FALSE - The transformation is not tried.

TEMP_DISABLE - This value has similar behavior as TRUE except that temporary table transformation is not tried.

The default value of the parameter is FALSE. You have to change the parameter value and create indexes on the joining columns of the fact table to take

advantage of this transformation.

Reference: Optimizer Transformations: Star Transformation