

# Oracle

## Exam 1z0-117

### Oracle Database 11g Release 2: SQL Tuning

Version: 7.0

[ Total Questions: 125 ]

**Question No : 1**

You instance has these parameter settings:

```
PARALLEL_DEGREE_POLICY=AUTO
PARALLEL_SERVERS_TARGET=64
PARALLEL_MIN_MINPERCENT=25
PARALLEL_MAX_SERVERS=128
PARALLEL_MIN_SERVERS=0
PARALLEL_MIN_TIME_THRESHOLD=10
PARALLEL_DEGREE_LIMIT=8
```

Which three statements are true about these settings if no hints are used in a SQL statement?

- A. A statement estimated for more than 10 seconds always has its degree of parallelism computed automatically.
- B. A statement with a computed degree of parallelism greater than 8 will be queued for a maximum of 10 seconds.
- C. A statement that executes for more than 10 seconds always has its degree of parallelism computed automatically.
- D. A statement with a computed degree of parallelism greater than 8 will raise an error.
- E. A statement with any computed degree of parallelism will be queued if the number of busy parallel execution processes exceeds 64.
- F. A statement with a computed degree of parallelism of 20 will be queued if the number of available parallel execution processes is less 5.

**Answer: C,E,F**

**Explanation:** C (not A): 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.

PARALLEL\_DEGREE\_LIMIT integer

A numeric value for this parameter specifies the maximum degree of parallelism the optimizer can choose for a SQL statement when automatic degree of parallelism is active. Automatic degree of parallelism is only enabled if PARALLEL\_DEGREE\_POLICY is set to AUTO or LIMITED.

E: `PARALLEL_SERVERS_TARGET` specifies the number of parallel server processes allowed to run parallel statements before statement queuing will be used. When the parameter `PARALLEL_DEGREE_POLICY` is set to `AUTO`, Oracle will queue SQL statements that require parallel execution, if the necessary parallel server processes are not available. Statement queuing will begin once the number of parallel server processes active on the system is equal to or greater than `PARALLEL_SERVER_TARGET`.

F: `PARALLEL_MIN_PERCENT`

`PARALLEL_MIN_PERCENT` operates in conjunction with `PARALLEL_MAX_SERVERS` and `PARALLEL_MIN_SERVERS`. It lets you specify the minimum percentage of parallel execution processes (of the value of `PARALLEL_MAX_SERVERS`) required for parallel execution. Setting this parameter ensures that parallel operations will not execute sequentially unless adequate resources are available. The default value of 0 means that no minimum percentage of processes has been set.

Consider the following settings:

`PARALLEL_MIN_PERCENT = 50`

`PARALLEL_MIN_SERVERS = 5`

`PARALLEL_MAX_SERVERS = 10`

If 8 of the 10 parallel execution processes are busy, only 2 processes are available. If you then request a query with a degree of parallelism of 8, the minimum 50% will not be met.

Note: With automatic degree of parallelism, Oracle automatically decides whether or not a statement should execute in parallel and what degree of parallelism the statement should use. The optimizer automatically determines the degree of parallelism for a statement based on the resource requirements of the statement. However, the optimizer will limit the degree of parallelism used to ensure parallel server processes do not flood the system. This limit is enforced by `PARALLEL_DEGREE_LIMIT`.

Values:

CPU

IO

integer

A numeric value for this parameter specifies the maximum degree of parallelism the optimizer can choose for a SQL statement when automatic degree of parallelism is active. Automatic degree of parallelism is only enabled if PARALLEL\_DEGREE\_POLICY is set to AUTO or LIMITED.

Reference:

PARALLEL\_MIN\_TIME\_THRESHOLD

PARALLEL\_DEGREE\_LIMIT

PARALELL\_MIN\_MINPERCENT

PARALELL\_SERVERS\_TARGET

### Question No : 2

You are administering a database, where an application frequently executes identical SQL statements with the same syntax.

How will you optimize the query results without retrieving data blocks from the storage?

- A. By setting the CURSOR\_SHARING parameter to FORCE.
- B. By using the bind variables and setting the CURSOR\_SHARING parameter to EXACT.
- C. By using the CACHE hint to pin the queries in the library cache
- D. By ensuring that RESULT\_CACHE\_MODE parameter is set to MANUAL and using the RESULT\_CACHE hint in the queries.
- E. By creating a SQL plan baseline for the identical statements.

### Answer: D

**Explanation:** As its name suggests, the query result cache is used to store the results of SQL queries for re-use in subsequent executions. By caching the results of queries, Oracle can avoid having to repeat the potentially time-consuming and intensive operations that generated the resultset in the first place (for example, sorting/aggregation, physical I/O, joins etc). The cache results themselves are available across the instance (i.e. for use by sessions other than the one that first executed the query) and are maintained by Oracle in a dedicated area of memory. Unlike our homegrown solutions using associative arrays or global temporary tables, the query result cache is completely transparent to our applications. It is also maintained for consistency automatically, unlike our own caching

programs.

Note:

RESULT\_CACHE\_MODE specifies when a ResultCache operator is spliced into a query's execution plan.

Values:

MANUAL

The ResultCache operator is added only when the query is annotated (that is, hints).

FORCE

The ResultCache operator is added to the root of all SELECT statements (provided that it is valid to do so).

For the FORCE setting, if the statement contains a NO\_RESULT\_CACHE hint, then the hint takes precedence over the parameter setting.

Incorrect:

A, B: CURSOR\_SHARING determines what kind of SQL statements can share the same cursors.

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.

C: The Oracle library cache is a component of the System Global Area (SGA) shared pool. Similarly to other Oracle cache structures, the point of the library cache is to reduce work – and therefore to improve performance – by caching the result of parsing and optimizing SQL or PL/SQL so that subsequent executions of the same SQL or PL/SQL require fewer preparatory steps to deliver a query result.

### Question No : 3

An application accessing your database got the following error in response to SQL query:

ORA-12827: insufficient parallel query slaves available

View the parallel parameters for your instance:

NAME	TYPE	VALUE
fast_start_parallel_rollback	string	LOW
parallel_adaptive_multi_tuning	boolean	TRUE
parallel_automatic_tuning	boolean	FALSE
parallel_degree_limit	string	32
parallel_degree_policy	string	LIMITED
parallel_execution_message_size	integer	16384
parallel_force_local	boolean	FALSE
parallel_io_cap_enabled	boolean	FALSE
parallel_max_servers	integer	128
parallel_min_servers	integer	50
parallel_server	Integer	0
parallel_server_instances	string	AUTO
parallel_server	boolean	1
parallel_servers_target	integer	8
parallel_threads_servers_per_cpu	integer	2

No hints are used and the session use default parallel settings.

What four changes could you make to help avoid the error and ensure that the query executes in parallel?

- A. Set PARELLEL\_DEGREE\_POLICY to AUTO.
- B. Increase the value of PARELLEL\_MAX\_SERVERS.
- C. Increase PARELLEL\_SERVERS\_TARGET.
- D. Decrease PARELLEL\_MIN\_PERCENT.
- E. Increase PARELLEL\_MIN\_SERVERS.
- F. Decrease PARELLEL\_MIN\_TIME\_THRESHOLD.
- G. Increase PARELLEL\_\_MIN\_TIME\_THRESHOLD.

**Answer: A,C,D,G**

**Explanation:** C: PARALLEL\_SERVERS\_TARGET specifies the number of parallel server processes allowed to run parallel statements before statement queuing will be used. When the parameter PARALLEL\_DEGREE\_POLICY is set to AUTO, Oracle will queue SQL statements that require parallel execution, if the necessary parallel server processes are not available. Statement queuing will begin once the number of parallel server processes active on the system is equal to or greater than PARALLEL\_SERVER\_TARGET.

By default, PARALLEL\_SERVER\_TARGET is set lower than the maximum number of parallel server processes allowed on the system (PARALLEL\_MAX\_SERVERS) to ensure each parallel statement will get all of the parallel server resources required and to prevent overloading the system with parallel server processes.

D:

Note: ORA-12827: insufficient parallel query slaves available

Cause: PARALLEL\_MIN\_PERCENT parameter was specified and fewer than minimum slaves were acquired

Action: either re-execute query with lower PARALLEL\_MIN\_PERCENT or wait until some running queries are completed, thus freeing up slaves

A, G: 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 No : 4**

You are administering database that supports an OLTP workloads. Most of the queries use an index range scan or index unique scan as access methods.

Which three scenarios can prevent the index access being used by the queries?

- A. When highly selective filters is applied on an indexed column of a table with sparsely populated blocks.
- B. When the rows are filtered with an IS NULL operator on the column with a unique key defined
- C. When the histogram statistics are not collected for the columns used in where clause.
- D. When a highly selective filter is applied on the indexed column and the index has very low value for clustering factor.
- E. When the statistics for the table are not current.

**Answer: A,B,E**

**Explanation:**

A: Low clustering factor promotes good performance.

The clustering\_factor measures how synchronized an index is with the data in a table. A table with a high clustering factor is out-of-sequence with the rows and large index range scans will consume lots of I/O. Conversely, an index with a low clustering\_factor is closely aligned with the table and related rows reside together of each data block, making indexes very desirable for optimal access.

Note:

\* Oracle SQL not using an index is a common complaint, and it's often because the optimizer thinks that a full-scan is cheaper than index access. Oracle not using an index can be due to:

\* (E) Bad/incomplete statistics – Make sure to re-analyze the table and index with dbms\_stats to ensure that the optimizer has good metadata.

\* Wrong optimizer\_mode – The first\_rows optimizer mode is to minimize response time, and it is more likely to use an index than the default all\_rows mode.

\* Bugs – See these important notes on optimizer changes in 10g that cause Oracle not to use an index.

\* Cost adjustment – In some cases, the optimizer will still not use an index, and you must decrease optimizer\_index\_cost\_adj.



**Question No : 5**

You are administering a database that supports an OLTP workload. Automatic optimizer statistics collection is scheduled in the night maintenance window. Some of the tables get updated frequently during day time and you notice a performance degradation for queries using those tables due to stale statistics.

Which two options might help to avoid the performance degradation of the queries?

- A. Set the global statistics preference STALE\_PERCENT to 0.
- B. Use dynamically sampling hint for the queries on frequently updated tables.
- C. Create histogram for the columns used frequently in the WHERE clause.
- D. Gather statistics with global statistics preference NO\_VALIDATE to TRUE.
- E. Set the OPTIMIZER\_USE\_PENDING\_STATISTICS parameter to TRUE.

**Answer: B,C**

**Explanation:** B: Dynamic sampling first became available in Oracle9i Database Release 2. It is the ability of the cost-based optimizer (CBO) to sample the tables a query references during a hard parse, to determine better default statistics for unanalyzed segments, and to verify its "guesses." This sampling takes place only at hard parse time and is used to dynamically generate better statistics for the optimizer to use, hence the name dynamic sampling.

There are two ways to use dynamic sampling:

The OPTIMIZER\_DYNAMIC\_SAMPLING parameter can be set at the database instance level and can also be overridden at the session level with the ALTER SESSION command. The DYNAMIC\_SAMPLING query hint can be added to specific queries.

C: A histogram is a collection of information about the distribution of values within a column. In some cases, the distribution of values within a column of a table will affect the optimizer's decision to use an index vs. perform a full-table scan. This scenario occurs when the value with a where clause has a disproportional amount of values, making a full-table scan cheaper than index access. Histograms are also important for determine the optimal table join order.

Incorrect:

A: Too much statistics would be gathered.

Note: 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%.

D: In Oracle PL/SQL, the **VALIDATE** keyword defines the *state of a constraint* on a column

in a table.

E: OPTIMIZER\_USE\_PENDING\_STATISTICS specifies whether or not the optimizer uses pending statistics when compiling SQL statements.

**Question No : 6**

You need to upgrade you Oracle Database 10g to 11g. You want to ensure that the same SQL plans that are currently in use in the 10g database are used in the upgraded database initially, but new, better plans are allowed subsequently.

Steps to accomplish the task:

1. Set the OPTIMIZER\_USE\_SQL\_BASELINE and OPTIMIZER\_CAPTURE\_SQL\_PLAN\_BASELINE to TRUE.
2. Bulk load the SQL Management Base as part of an upgrade using an STS containing the plans captured in Oracle Database 10g.
3. Evolve the plan baseline using the DBMS\_SPM.EVOLVE\_PLAN\_BASELINE procedure.
4. Fix the plan baseline – using the DBMS\_SPM.ALTER\_SQL\_PLANBASELINE procedure.
5. Accept new, better plans using the DBMS\_SPM.ALTER\_SQL\_PLAN\_BASELINE procedure and manually load them to the existing baseline.
6. Set OPTIMIZER\_CAPTURE\_SQL\_PLAN\_BASELINES to FALSE.

Identify the required steps.

- A. 1, 3, 4, 5
- B. 1, 6, 3, 4, 5
- C. 1, 2, 3, 5
- D. 1, 2, 3, 4
- E. 1, 6, 3
- F. 1 and 2

**Answer: F**

**Explanation:** \* (1) OPTIMIZER\_CAPTURE\_SQL\_PLAN\_BASELINES

In Oracle Database 11g a new feature called SQL Plan Management (SPM) has been