

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

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



QUESTION 1

Which four types of column filtering may benefit from partition pruning when accessing tables via partitioned indexes?

- A. Equality operates on List-Partitioned Indexes
- B. Not Equal operates on a Global Hash-Partitioned Indexes
- C. Equality operates on System-Partitioned Tables
- D. In-List operates on Range-Partitioned Indexes
- E. Not Equal operates on a local Hash-Partitioned Indexes
- F. Equality operates on Range-Partitioned Indexes
- G. Equality operates on Hash-Partitioned Indexes

Correct Answer: ADFG

Oracle Database prunes partitions when you use range, LIKE, equality (A, F), and IN-list (D) predicates on the range or list partitioning columns, and when you use equality (G) and IN-list predicates on the hash partitioning columns.

Reference: Oracle Database VLDB and Partitioning Guide 11g, Information that can be Used for Partition Pruning

QUESTION 2

Which two tasks are performed during the optimization stage of a SQL statement?

- A. Evaluating the expressions and conditions in the query
- B. Checking the syntax and analyzing the semantics of the statement
- C. Separating the clauses of the SQL statement into structures that can be processed
- D. Inspecting the integrity constraints and optimizing the query based on this metadata
- E. Gathering the statistics before creating the execution plan for the statement

Correct Answer: DE

Note:

Oracle SQL is parsed before execution, and a hard parse includes these steps: *

1.
Loading into shared pool - The SQL source code is loaded into RAM for parsing. (the "hard" parse step)
2.
Syntax parse - Oracle parses the syntax to check for misspelled SQL keywords.

3.

Semantic parse - Oracle verifies all table and column names from the dictionary and checks to see if you are authorized to see the data.

4.

Query Transformation - If enabled (query_rewrite=true), Oracle will transform complex SQL into simpler, equivalent forms and replace aggregations with

materialized views, as appropriate.

5.

Optimization - Oracle then creates an execution plan, based on your schema statistics (or maybe with statistics from dynamic sampling in 10g).

6.

Create executable - Oracle builds an executable file with native file calls to service the SQL query.

*

The parsing process performs two main functions:

o Syntax Check: is the statement a valid one. Does it make sense given the SQL grammar documented in the SQL Reference Manual. Does it follow all of the

rules for SQL.

o Semantic Analysis: Going beyond the syntax ? is the statement valid in light of the objects in the database (do the tables and columns referenced exist). Do you

have access to the objects ? are the proper privileges in place? Are there ambiguities in the statement ? for example if there are two tables T1 and T2 and both

have a column X, the query ?select X from T1, T2 where ?? is ambiguous, we don't know which table to get X from. And so on.

So, you can think of parsing as basically a two step process, that of a syntax check to check the validity of the statement and that of a semantic check ? to ensure

the statement can execute properly.

Reference: Oracle hard-parse vs. soft parse

QUESTION 3

See the code fragment:

```
PARALLEL_DEGREE_POLICY=MANUAL
PARALLEL_MIN_PERCENT = 50
PARALLEL_MAX_SERVERS = 128
PARALLEL_MIN_SERVERS = 0
PARALLEL_DEGREE_LIMIT = 8
```

You execute the following query:

```
SELECT /*+ FULL (c, 8) */ c.cust_last_name, s.time_id, s.quantity_id,
s.quantity_sold
FROM oe.customers c, sh.sales s
WHERE s.cust_id = c.customer_id;
```

You receive the following error message:

```
ORA-12827: insufficient parallel query slaves available
```

Which three parameter settings could you change to avoid this error?

- A. Decrease the value of PARALLEL_MIN_PERCENT
- B. Increase the value of PARALLEL_MAX_SERVERS
- C. Increase the value of PARALLEL_MIN_SERVERS
- D. Reduce the value of PARALLEL_MIN_TIME_THRESHOLD
- E. Increase the value of PARALLEL_DEGREE_LIMIT
- F. Set the PARALLEL_DEGREE_POLICY = AUTO
- G. Set the PARALLEL_DEGREE_POLICY = LIMITED

Correct Answer: ABG

A: 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

B: Your query doesn't run because you've told Oracle not to run it unless at least 5% of the parallel execution processes are available for your query.

Set PARALLEL_MIN_PERCENT=0 or increase the number of parallel execution processes by increasing the PARALLEL_MAX_SERVERS parameter.

G: PARALLEL_DEGREE_POLICY

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

LIMITED

Enables automatic degree of parallelism for some statements but statement queuing and in-memory Parallel Execution are disabled. Automatic degree of parallelism is only applied to those statements that access tables or indexes decorated explicitly with the PARALLEL clause. Tables and indexes that have a degree of parallelism specified will use that degree of parallelism.

Note: 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.

You can use this parameter in conjunction with PARALLEL_ADAPTIVE_MULTI_USER. In a multi-user environment, an individual user or application can set PARALLEL_MIN_PERCENT to a minimum value until sufficient resources are available on the system and an acceptable degree of parallelism is returned.

QUESTION 4

A database supports three applications: CRM, ERP, and ACC. These applications connect to the database by using three different services: CRM_SRV for the

CRM application, ERP_SRV for the ERP application, and ACC_SRV for the ACC application.

You enable tracing for the ACC_SRV service by issuing the following command:

SQL> EXECUTE DBMS for the ACC_SRV service by issuing the following command:

```
SQL> EXECUTIVE DBMS_MONITOR. SERV_MOD_ACT_TRACE_ENABLE
```

```
(service_name => 'ACC_SRV\$', waits => TRUE, binds => FALSE, instance_name => 'inst1\');
```

Which statement is true?

- A. All trace information for the service connection to inst1 will be stored in a single trace file.
- B. A trace file is not created because the module name is not specified.
- C. A single trace file is created for each session that uses the ACC_SRV service.
- D. Only those SQL statements that are identified with the ACC_SRV service executed on the inst1 instance are recorded in trace files.
- E. All trace information for the ACC_SRV service connected to inst1 is stored in multiple trace files, which can be consolidated by using the tkprof utility.

Correct Answer: C

```
SERV_MOD_ACT_TRACE_ENABLE
```

serv_mod_act_trace_enable and serv_mod_act_trace_disable, which enables and disables trace for given service_name, module and action.

For example for a given service name you can trace all session started from SQL*Plus. Module and action in your own created application can be set using

dbms_application_info set_module and set_action procedures.

serv_mod_act_trace_enable fills sys table wri\$_tracing_enabled and view dba_enabled_traces on top of this table as follows:

```
SQL> exec dbms_monitor.serv_mod_act_trace_enable(service_name=>'orcl', module_name=>'SQL*Plus')
```

PL/SQL procedure successfully completed.

```
SQL> select * from sys.wri$_tracing_enabled;
```

```
TRACE_TYPE PRIMARY_ID QUALIFIER_ID1 QUALIFIER_ID2 INSTANCE_NAME
FLAGS
```

```
----- 4 orcl SQL*Plus 8
```

```
SQL> select * from dba_enabled_traces;
```

```
TRACE_TYPE PRIMARY_ID QUALIFIER_ID1 QUALIFIER_ID2 WAITS BINDS
INSTANCE_NAME
```

```
----- SERVICE_MODULE orcl SQL*Plus TRUE FALSE
```

QUESTION 5

Which three statements are true the Automatic Tuning Optimizer (ATO)?

- A. It identifies the objects with stale or missing statistics and gathers statistics automatically.
- B. It investigates the effect of new or modified indexes on the access paths for a workload and recommends running that statistics through the SQL Access Advisor.
- C. It recommends a SQL profile to help create a better execution plan.
- D. It picks up resource-intensive SQL statements from the ADDM and recommends the use of materialized views to improve query performance.
- E. It identifies the syntactic, semantic, or design problems with structure of SQL statements leading to poor performance and suggests restricting the statements.
- F. It identifies resource-intensive SQL statements, runs them through the SQL Tuning Advisor, and implements the recommendations automatically.

Correct Answer: ADF

Under tuning mode, the optimizer can take several minutes to tune a single statement. It is both time and resource intensive to invoke Automatic Tuning Optimizer every time a query must be hard-parsed. Automatic Tuning Optimizer is

meant for complex and high-load SQL statements that have nontrivial impact on the database.

Automatic Database Diagnostic Monitor (ADDM) proactively identifies high-load SQL statements that are good candidates for SQL tuning. The automatic SQL tuning feature also automatically identifies problematic SQL statements and implements tuning recommendations during system maintenance windows as an automated maintenance task.

The Automatic Tuning Optimizer performs the following types of tuning analysis:

Statistics Analysis SQL Profiling Access Path Analysis SQL Structure Analysis Alternative Plan Analysis

Note:

* Oracle Database uses the optimizer to generate the execution plans for submitted SQL statements. The optimizer operates in the following modes:

Normal mode The optimizer compiles the SQL and generates an execution plan. The normal mode generates a reasonable plan for the vast majority of SQL statements. Under normal mode, the optimizer operates with very strict time constraints, usually a fraction of a second.

Tuning mode

The optimizer performs additional analysis to check whether it can further improve the plan produced in normal mode. The optimizer output is not an execution plan, but a series of actions, along with their rationale and expected benefit for producing a significantly better plan. When running in tuning mode, the optimizer is known as the Automatic Tuning Optimizer.

QUESTION 6

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.

Correct Answer: BC

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 7

Examine the following query and execution plan: Which query transformation technique is used in this scenario?

Examine the following query and execution plan:

```
SQL> SELECT prod_id prod_name
      FROM producers p
      Where prod_list_prices > (SELECT MAX(unit_cost)
      FROM costs c
      WHERE p.prod_id=c.prod_id);
```

Execution Plan

Plan hash value: 2297807746

Pstart	id Pstop	Operation	Name	Rows	Bytes	Cost	(%CPU)	Time
	0	SELECT STATEMENT		4	140	82	(8)	00:00:01
*	1	HASH JOIN		4	140	82	(8)	00:00:01
	2	VIEW	VW_SQL_1	72	1872	78	(7)	00:00:01
	3	HASH GROUP BY		72	648	78	(7)	00:00:01
	4	PARTITION RANGE ALL		82112	721K	74	(2)	00:00:01
1	28							
	5	TABLE ACCESS FULL	COSTS	82112	721K	74	(2)	00:00:01
1		28						
	6	TABLE ACCESS FULL	PRODUCTS	72	648	3	(0)	00:00:01

Predicate Information (Identified by operation id);

```
1- access ("P", "PROD_ID"=by operation id):
  Filter ("PROD_LIST_PRICE"> "MAX(UNIT_COST)")
```

- A. Join predicate push-down
- B. Subquery factoring
- C. Subquery unnesting
- D. Join conversion

Correct Answer: A

*

Normally, a view cannot be joined with an index-based nested loop (i.e., index access) join, since a view, in contrast with a base table, does not have an index defined on it. A view can only be joined with other tables using three methods: hash, nested loop, and sort-merge joins.

*

The following shows the types of views on which join predicate pushdown is currently supported. UNION ALL/UNION view Outer-joined view Anti-joined view Semi-joined view DISTINCT view GROUP-BY view

QUESTION 8

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.

Correct Answer: CEF

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: PARALELL_MIN_MINPERCENT

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 PARALLEL_MIN_MINPERCENT
PARALLEL_SERVERS_TARGET

QUESTION 9

Examine the Exhibit and view the query and its execution plan.

```
SQL>EXPLAIN PLAN FOR
SELECT /*+ PARALLEL (4) */ customers.cust_first_name, customers.cust_last_name,
MAX (QUANTITY_SOLD), AVG (QUANTITY_SOLD)
FROM sales, customers
WHERE sales, customers
WHERE sales.cust_id=customers.cust_id
GROUP By customers.cust_first_name, customers.cust_last_name;
```

Explained

PLAN_TABLE_OUTPUT

Plan hash value: 4060011603

Id	Operation	Name	Rows	Bytes	TQ	IN-OUT	PQ	Distrib
0	SELECT STATEMENT		925	25900				
1	PX COORDINATOR							
2	PX SEND QC (RANDOM)	:TQ10003	925	25900	Q1,03	P->S	QC	RAND
3	HASH GROUP BY		925	25900	Q1,03	PCWP		
4	PX RECEIVE		925	25900	Q1,03	PCWP		
5	PX SEND HASH	:TQ10002	925	25900	Q1,02	P->P	HASH	
6	HASH JOIN BUFFERED		925	25900	Q1,02	PCWP		
7	PX RECEIVE		630	12600	Q1,02	PCWP		
8	PX SEND HASH	:TQ10000	630	12600	Q1,00	P->P	PCWP	
9	PX BLOCK ITERATOR		630	12600	Q1,00	PCWP		
10	TABLE ACCESS FULL	CUSTOMERS	630	12600	Q1,00	PCWP		
11	PX RECEIVE		960	7680	Q1,02	PCWP		
12	PX SEND HASH	:TQ10001	960	7680	Q1,01	P->P	HASH	
13	PX BLOCK ITERATOR		960	7680	Q1,01	PCWC		
14	TABLE ACCESS FULL	SALES	960	7680	Q1,01			

Predicate Information (identified by operation id):
6 - access ("SALES", "CUST_ID"= "CUSTOMERS", "CUST_ID")

(PLAN)			
3	HASH GROUP BY		
4	PX RECEIVE		
5	PX SEND HASH	:TQ10002	
6	HASH JOIN BUFFERED		
7	PX RECEIVE		
8	PX SEND HASH	:TQ10000	
9	PX BLOCK ITERATOR		
10	TABLE ACCESS FULL	CUSTOMERS	
11	PX RECEIVE		
12	PX SEND HASH	:TQ10001	
13	PX BLOCK ITERATOR		
14	TABLE ACCESS FULL	SALES	

Which statement is correct about the parallel executions plan?

- A. The CUSTOMERS and SALES tables are scanned simultaneously in parallel and then joined in parallel.
- B. First, the CUSTOMERS table is scanned in parallel, then the SALES table is scanned in parallel, and then they are joined serially.
- C. First, the SALES table is scanned in parallel, then the CUSTOMERS table is scanned in parallel, and then they are joined in parallel.
- D. The CUSTOMERS and SALES tables are scanned simultaneously in parallel and then joined serially.
- E. First, the CUSTOMERS table is scanned in parallel, then the SALES table is scanned in parallel, and then they are joined in parallel.

Correct Answer: A

As per exhibit:

Line 7 and line 11 are run in parallel.

Line 8 and line 12 are run in parallel.

Line 9 and line 13 are run in parallel.

Line 10 and line 14 are run in parallel.

Line 6 is a PCWP (parallel combined with parent) and the parent is a P-> P (Parallel to parallel) operation.

QUESTION 10

You are administering a database that supports an OLTP workload in which one of the applications inserts rows in a table until 12 noon every, after which multiple users perform frequent queries on the table. You want the statistics to be more representative of the table population.

What must be done to ensure that an optimizer uses the latest statistics of the table?

- A. Set the STALE_PERCENT preference to 0.
- B. Set the OPTIMIZER_MODE parameter to ALL_ROWS.
- C. Set the OPTIMIZER_DYNAMIC_SAMPLING parameter to 0.
- D. Use the FIRST_ROWS_n hint in the queries.
- E. Unlock and gather statistics for the table after inserts are done and lock them again.

Correct Answer: E

*

For tables that are substantially modified in batch operations, such as with bulk loads, gather statistics on these tables as part of the batch operation. Call the DBMS_STATS procedure as soon as the load operation completes.

* Statistics for a table or schema can be locked. After statistics are locked, you can make no modifications to the statistics until the statistics have been unlocked. Locking procedures are useful in a static environment in which you want to guarantee that the statistics never change.

The DBMS_STATS package provides two procedures for locking (LOCK_SCHEMA_STATS and LOCK_TABLE_STATS) and two procedures for unlocking statistics (UNLOCK_SCHEMA_STATS and UNLOCK_TABLE_STATS).

Incorrect:

A: STALE_PERCENT cannot be set to 0.

* 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%.

B: Optimizer_mode applies to the database, not to a specific table.

*

Possible values for optimizer_mode = choose/ all_rows/ first_rows/ first_rows[n]

Important facts about ALL_ROWS

1.

ALL_ROWS considers both index scan and full scan and based on their contribution to the overall query, it uses them. If Selectivity of a column is low, optimizer may use index to fetch the data (for example `where employee_code=7712\`), but if selectivity of column is quite high (\`where deptno=10\`), optimizer may consider doing Full table scan. With ALL_ROWS, optimizer has more freedom to its job at its best.

2.

Good for OLAP system, where work happens in batches/procedures. (While some of the report may still use FIRST_ROWS depending upon the anxiety level of report reviewers)

3.

Likes hash joins over nested loop for larger data sets.

C: Optimizer dynamic sampling refers to the ability of the SQL optimizer to take a sample of rows from a table to calculate missing statistics. Dynamic sampling can be controlled with the OPTIMIZER_DYNAMIC_SAMPLING parameter or the DYNAMIC_SAMPLING hint.

level 0 - do not use dynamic sampling

D: First_row_n cannot be used in this way.

QUESTION 11

Examine the statements being executed for the first time:

```
SQL SELECT last_name, email, salary, FROM employees WHERE employees_id = 100;
```

```
SQL> SELECT last_name, email, salary FROM employees WHERE employees_id = 110;
```

```
SQL > SELECT last_name, email, salary FROM employees WHERE employees_id = 130;
```

Steps followed by a SQL statement during parsing:

1.

Search for a similar statement in the shared pool.

2.

Search for an identical statement in the shared pool.

3.

Search the SQL area of identical statement already in the shared pool.

4.

Proceed through the remaining steps of the parse phase to ensure that the execution plan of the existing statements is applicable to the view statement.

5.

Perform hard parsing.

6.

Share the SQL area of the similar statement already in the shared pool.

Identify the required steps in the correct sequence used by the third query.

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

Correct Answer: F

2) before 5).

Note:

* When application code is run, Oracle attempts to reuse existing code if it has been executed previously and can be shared. If the parsed representation of the statement does exist in the library cache and it can be shared, then Oracle reuses the existing code. This is known as a soft parse, or a library cache hit. If Oracle is unable to use existing code, then a new executable version of the application code must be built. This is known as a hard parse, or a library cache miss.

Reference: Oracle Database Performance Tuning Guide, SQL Sharing Criteria

QUESTION 12

You executed the following statements:

```
SQL> SET AUTOTRACE TRACEONLY EXPLAIN STAT
```

```
SQL> SELECT e.name, e.sal, d.name  
DEOM emp, dept d  
WHERE e.dept_id = d.dept_id;
```

Which two statements are true about the query execution?

- A. The execution plan is generated and fetched from the library cache.
- B. The query executes and displays the execution plan and statistics.
- C. The query executes and inserts the execution plan in PLAN_TABLE.
- D. The query executes and execution plan is stored in the library cache and can be viewed using v\$SQL_PLAN.
- E. The query will always use the plan displayed by the AUTOTRACE output.

Correct Answer: BD

B: set autotrace traceonly: Displays the execution plan and the statistics (as set autotrace on does), but doesn't print a query's result.

Note:

/ Autotrace

Autotrace can be configured to run the SQL and gives a plan and statistics afterwards or just give you an explain plan without executing the query. To achieve this

use the following:

*

Explain only

set autotrace traceonly explain

*

Execute with stats and explain plan

set autotrace on explain stat (with data returned by query)

or

autotrace traceo expl stat (without data returned by query)

*

To make the output from an autotrace more readable

col plan_plus_exp format a100

*

Turn off autotrace

set autotrace off

/ V\$SQL_PLAN contains the execution plan information for each child cursor loaded in the library cache.
http://docs.oracle.com/cd/E11882_01/server.112/e40402/dynviews_3054.htm#REFRN30250

QUESTION 13

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 14

An application frequently executed similar types of queries that vary only in the usage of literals in the WHERE clause. You plan to use bind variable in place of literal values.

The CURSOR_SHARING parameter to set to EXACT.

Which two statements are true about the usage of bind variables?

- A. The number of latch gets in the library cache will be reduced.
- B. Bind peeking will take place and subsequent execution of queries can have different plans based on the cardinality of the bind value in the column.
- C. Bind peeking will take place and subsequent execution of queries can have different plans only when the histograms exists on a column used in WHERE clause.
- D. Bind peeking will not happen and the optimizer will use the same plan for all bind values if no histograms exist on a column use in the WHERE clause.
- E. Bind peeking will happen and subsequent execution of queries will have the same parent cursor but different child cursors.

Correct Answer: AB

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

EXACT

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

Note:

EXACT-This is the default setting. With this value in place, the query is not rewritten to use bind variables.

With CURSOR_SHARING=EXACT (the default), every unique SQL statement executed will create a new entry in V\$SQL, it will be hard-parsed, and an execution plan will be created just for it. There can be hundreds or thousands of very similar queries in the shared pool that differ only in the literals used in the SQL statement itself. This implies that the database is forced to hard-parse virtually every query, which, in turn, not only consumes a lot of CPU cycles but also leads to decreased scalability.

The database just cannot hard-parse hundreds or thousands of SQL statements concurrently-- the application ends up waiting for the shared pool to become available. One of the major scalability inhibitors in the database is not using bind variables. That was the motivation behind adding CURSOR_SHARING=FORCE .

QUESTION 15

Examine Exhibit 1 to view the query and its execution plan.

SQL> select

```
First_name, e.department_id, d.department_id, d.department_name from employees e, departments d
Where e.department_id = d.department_id and last_name like '%a%'
```

106 rows selected.

Execution plan

Plan hash value: 1473400139

Id	Operation	Name	Rows	Bytes	Cost	(%CPU)	Time
0	SELECT STATEMENT		106	2576	6	(34)	00:00:01
1	MERGE JOIN		106	2756	6	(34)	00:00:01
2	TABLE ACCESS BY INDEX ROWID	DEPARTMENTS	27	432	2	(0)	00:00:01
3	INDEX FULL SCAN	DEPT_IDPK	27		1	(0)	00:00:01
4	SORT JOIN		107	1070	4	(50)	00:00:01
5	VIEW	indes\$_join\$_001	107	7070	3	(34)	00:00:01
6	HASH JOIN						
7	INDEX FAST FULL SCAN	EMP_DEPARTMENT_IX	107	1070	1	(0)	00:00:01
8	INDEX FAST FULL SCAN	EMP_NAME_IX	107	1070	1	(0)	00:00:01

Predicate information (Identified by Operation id):

```
4 - access ("E", "DEPARTMENT_ID" = "D", DEPARTMENT_ID)
   - filter ("E", "DEPARTMENT_ID" = "D", DEPARTMENT_ID)
6 - access (ROWID=ROWID)
8 - filter ("LAST_NAME" LIKE '%A%')
```

Statistics

```
1 recursive calls
0 db block gets
20 consistent gets
0 physical reads
0 redo size
4034 bytes sent via SQL*NET to client
596 bytes received via SQL*NET from client
9 SQL*NET roundtrips to/from client
1 sorts (memory)
0 sorts (disk)
106 rows proceed
```

Examine Exhibit 2 to view the structure and indexes for the EMPLOYEES and DEPARTMENTS tables. Examine Exhibit 3 to view the initialization parameters for the instance.

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR(20)
LAST_NAME	NOT NULL	VARCHAR(25)
EMAIL	NOT NULL	VARCHAR (25)
PHONE_NUMBER		VARCHAR(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR (10)
SALARY		NUMBER (8, 2)
COMISSION_PCT		NUMBER (2, 2)
MANAGER_ID		NUMBER (6)
DEPARTMENT_ID		NUMBER (4)

INDEX_NAME	INDEX_TYPE	COLUMN_NAME
EMP_NAME_IX	NORMAL	LAST_NAME
EMP_MANAGER_IX	NORMAL	MANAGER_ID
EMP_JOB_IX	NORMAL	JOB_ID
EMP_DEPARTMENT_IX	NORMAL	DEPARTMENT_ID
EMP_EMP_ID_PK	NORMAL	EMPLOYEE_ID
EMP_EMAIL_UK	NORMAL	EMAIL

Departments

Name	Null?	Type
DEPARTMENT	NOT NULL	NUMBER (4)
DEPARTMENT_NAME	NOT NULL	VARCHAR (30)
MANAGER_ID		NUMBER (6)
LOCATION_ID		NUMBER (4)

INDEX_NAME	INDEX_TYPE	COLUMN_NAME
DEPT_LOCATION_IX	NORMAL	LOCATION_ID
DEPT_ID_PK	NORMAL	DEPARTMENT_ID

NAME	TYPE	VALUE
optimizer_capture_sql_plan_baselines	boolean	FALSE
optimizer_dynamic_sampling	integer	2
optimizer_features_sampling	string	11.2.0.1
optimizer_index_catching	integer	0
optimizer_index_cost_adj	integer	100
optimizer_mode	string	ALL_ROWS
optimizer_secure_view_merging	boolean	TRUE
optimizer_use_invisible_indexes	boolean	FALSE
optimizer_use_pending_statistics	boolean	FALSE
optimizer_use_sql_plan_baselines	boolean	TRUE

Why is sort-merge join chosen as the access method?

- A. Because the OPTIMIZER_MODE parameter is set to ALL_ROWS.
- B. Because of an inequality condition.
- C. Because the data is not sorted in the LAST_NAME column of the EMPLOYEES table
- D. Because of the LIKE operator used in the query to filter out records

Correct Answer: A

Incorrect:

- B: There is not an inequality condition in the statement.
- C: Merge joins are beneficial if the columns are sorted.
- D: All regular joins should be able to use Hash or Sort Merge, except LIKE, !=, and NOT ... joins.

Note:

*

A sort merge join is a join optimization method where two tables are sorted and then joined.

*

A "sort merge" join is performed by sorting the two data sets to be joined according to the join keys and then merging them together. The merge is very cheap, but the sort can be prohibitively expensive especially if the sort spills to disk. The cost of the sort can be lowered if one of the data sets can be accessed in sorted order via an index, although accessing a high proportion of blocks of a table via an index scan can also be very expensive in comparison to a full table scan.

*

Sort merge joins are useful when the join condition between two tables is an inequality condition (but not a nonequality) like =. Sort merge joins

perform better than nested loop joins for large data sets. You cannot use hash joins unless there is an equality

condition.

*

When the Optimizer Uses Sort Merge Joins

The optimizer can choose a sort merge join over a hash join for joining large amounts of data if any of the following conditions are true:

/ The join condition between two tables is not an equi-join.

/ Because of sorts already required by other operations, the optimizer finds it is cheaper to use a sort merge than a hash join. Reference: Oracle Database Performance Tuning Guide , Sort Merge Joins

[1Z0-117 Study Guide](#)

[1Z0-117 Exam Questions](#)

[1Z0-117 Braindumps](#)