

## 1Z0-117<sup>Q&As</sup>

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**

Examine the Exhibit.

```
CREATE TABLE dept AS SELECT * FROM departments;
ALTER TABLE dept PARALLEL 2;

CREATE TABLE emp_range_did PARTITION BY RANGE (department_id)
(PARTITION emp_p1 VALUES LESS THAN (150),
PARTITION emp_p5 VALUES LESS THAN (MAXVALUE) )
AS SELECT * FROM employees;

ALTER TABLE emp_range_did PARALLEL 2;

EXPLAIN PLAN FOR
SELECT /*PQ_DISTRIBUTE (d NONE PARTITION) ORDERED */ e.last_name, d.department_name
FROM emp_range_did e, dept d
WHERE e.department_id = d.department_id;
```

Id	Operations	Name	Rows	Bytes	Cost	Pstart	Pstop
0	SELECT STATEMENT		284	16188	6		
1	PX COORDINATOR						
2	PX SEND QC (RANDOM) :	TQ10001	284	16188	6		
3	HASH JOIN		284	16188	6		
4	PX PARTITION RANGE ALL		284	7668	2	1	2
5	TABLE ACCESS FULL	EMP_RANGE_DID	284	7668	2	1	2
6	BUFFER SORT						
7	PX RECEIVE		21	630	2		
8	PX SEND PARTITION (KEY) :	TQ10000	21	630	2		
9	TABLE ACCESS FULL	DEPT	21	630	2		

Which two options are true about the execution plan and the set of statements?

- A. The query uses a partial partition-wise join.
- B. The degree of parallelism is limited to the number of partitions in the EMP\_RANGE\_DID table.
- C. The DEPT table id dynamically distributed based on the partition keys of the EMP\_RANGE\_DID table.
- D. The server process serially scans the entire DEPT table for each range partition on the EMP\_RANGE\_DID table.
- E. The query uses a full partition-wise join.

Correct Answer: AD

**QUESTION 2**

View the exhibit and examine the findings provided by the SQL Tuning Advisor for SELECT Statement.

**2- Alternative Plan Finding**

Some alternate execution plans for this statement were found by searching the system's real-time and historical performance data.

The following table lists these plans ranked by their average elapsed.

See section "ALTERNATIVE PLANS SECTION" for detailed information on each plan.

Id	plan	hash last seen	elapsed (s)	origin	note
1	1378942017	2011-02-05/22:12:08	0.000	Cursor Cache	Original plan
2	2842999589	2011-02-05/22:12:05	0.002	STS	

**Information**

The original plan appears to have the best performance, based on the elapsed time per execution. However, if you know that one alternative plan is better than Original Plan, you can create a SQL plan baseline for it. This will instruct the Oracle optimizer to pick it over any other choices in the future.

```
Execute dbms_sqltune.create_sql_plan_baseline (task_name=> 'TASK_XXXXX', object_id=> 2, task_owner=> 'SYS', hash=> XXXXXXXXX);
```

A SWL plan baseline already exists for the execution plan.

What two methods can you use to ensure that an alternate plan becomes an accepted plan?

- A. Use the DBMS\_SPM.ALTER\_SQL\_PLAN\_BASELINE function.
- B. Use the DBMS\_SQLTUNE.CREATE\_SQL\_PLAN\_BASELINE function.
- C. Use the DBMS\_SQLTUNE.CREATE\_SQL\_PLAN\_BASELINE function and run the DBMS\_STATS to manually refresh stale statistics.
- D. Use the DBMS\_SPM.LOAD\_PLANS\_FROM\_SQLSET function.

Correct Answer: CD

C: To adopt an alternative plan regardless of whether SQL Tuning Advisor recommends it, call DBMS\_SQLTUNE.CREATE\_SQL\_PLAN\_BASELINE. You can use this procedure to create a SQL plan baseline on any existing reproducible plan.

D: LOAD\_PLANS\_FROM\_SQLSET Function

This function loads plans stored in a SQL tuning set (STS) into SQL plan baselines. The plans loaded from STS are not verified for performance but added as

accepted plans to existing or new SQL plan baselines. This procedure can be used to seed SQL management base with new SQL plan baselines.

Note:

\* While tuning a SQL statement, SQL Tuning Advisor searches real-time and historical performance data for alternative execution plans for the statement. If plans

other than the original plan exist, then SQL Tuning Advisor reports an alternative plan finding.

SQL Tuning Advisor validates the alternative execution plans and notes any plans that are not reproducible. When reproducible alternative plans are found, you

can create a SQL plan baseline to instruct the optimizer to choose these plans in the future.

Incorrect:

Not A: 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.

Usage Notes

When a single plan is specified, one of various statuses, or plan name, or description can be altered. When all plans for a SQL statement are specified, one of

various statuses, or description can be altered. This function can be called numerous times, each time setting a different plan attribute of same plan(s) or different

plan(s).

Reference: Oracle Database Performance Tuning Guide, Alternative Plan Analysis

---

### QUESTION 3

How can you reduce fragmentation of an index without affecting the current transactions that are using the index?

- A. Use the ANALYZE INDEX . . . command
- B. Use the ALTER INDEX . . . VALIDATE STRUCTURE command
- C. Us the ALTER INDEX . . . REBUILD ONLINE command
- D. Use the ALTER INDEX . . . DEALLOCATE UNUSED command

Correct Answer: D

Use the deallocate\_unused\_clause to explicitly deallocate unused space at the end of the index and make the freed space available for other segments in the tablespace.

If index is range-partitioned or hash-partitioned, then Oracle Database deallocates unused space from each index partition. If index is a local index on a composite-partitioned table, then Oracle Database deallocates unused space from each index subpartition.

Reference: Oracle Database SQL Language Reference 11g, alter index

---

### QUESTION 4

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.

Correct Answer: B

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 5

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 be 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 Base, 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.

[1Z0-117 VCE Dumps](#)

[1Z0-117 Study Guide](#)

[1Z0-117 Braindumps](#)