1Z0-117^Q&As^

Oracle Database 11g Release 2: SQL Tuning Exam

Pass Oracle 1Z0-117 Exam with 100% Guarantee

Following Questions and Answers are all new published by Oracle Official Exam Center

QUESTION 1

You plan to bulk load data using INSERT /*+PARALLEL*/ INTO . . . . SELECT FROM statements.

Which four types of operations can execute in parallel on tables that have no bitmapped indexes or materialized views defined on term?

A. Direct path insert of a million rows into a partitioned, index-organized table containing one million rows.

B. Direct path insert of a million rows into a partitioned, index-organized table containing 10 million rows.

C. Direct path insert of a million rows into a nonpartitioned, index-organized table containing one million rows.

D. Direct path insert of a million rows into a nonpartitioned, heap-organized table containing 10 million rows.

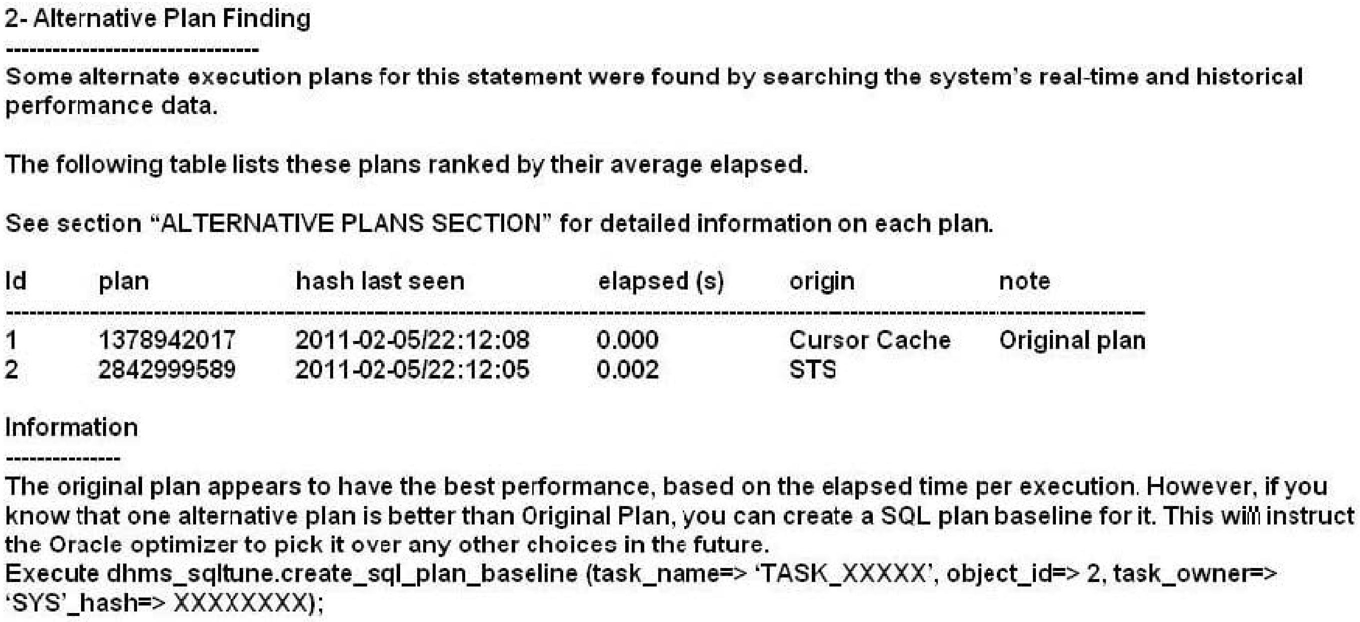
E. Direct path insert of a million rows into a nonpartitioned, heap-organized table containing one million rows.

Correct Answer: ABDE

Direct-path INSERT is not supported for an index-organized table (IOT) if it is not partitioned, if it has a mapping table, or if it is reference by a materialized view.

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 dhms_sqltune.create_sql_plan_baseline (task_name=> 'TASK_XXXXX', object_id=> 2, task_owner=>
'SYS'_hash=> XXXXXXXX);
```

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

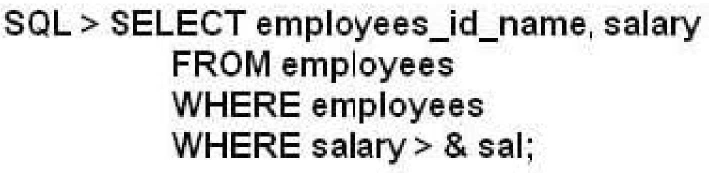
various statuses, or description can be altered. This function can be called numerous times, each time setting a different plan attribute of same plan(s) or different

plan(s).

Reference: Oracle Database Performance Tuning Guide, Alternative Plan Analysis

QUESTION 3

You execute the following query for the first time:



```
SQL > SELECT employees_id_name, salary
             FROM employees
             WHERE employees
             WHERE salary > & sal;
```

Examine the SQL statement processing steps:

1.

The value of the variable SAL is obtained to run the query.

2.

The syntax of the query is checked

3.

A parse tree for the query is generated

4.

Semantics for the query are checked

5.

The required rows are fetched

6.

The SQL is executed to produce the required result.

Which is the correct order of execution of the above query?

A. 1, 2 3, 4, 5, 6

B. 1, 4, 2, 3, 6, 5

C. 2, 4, 1, 3, 6, 5

D. 2, 3, 1, 4, 6, 5

Correct Answer: C

Step 1: Syntax check (2)

Step 2: Semantic check (4)

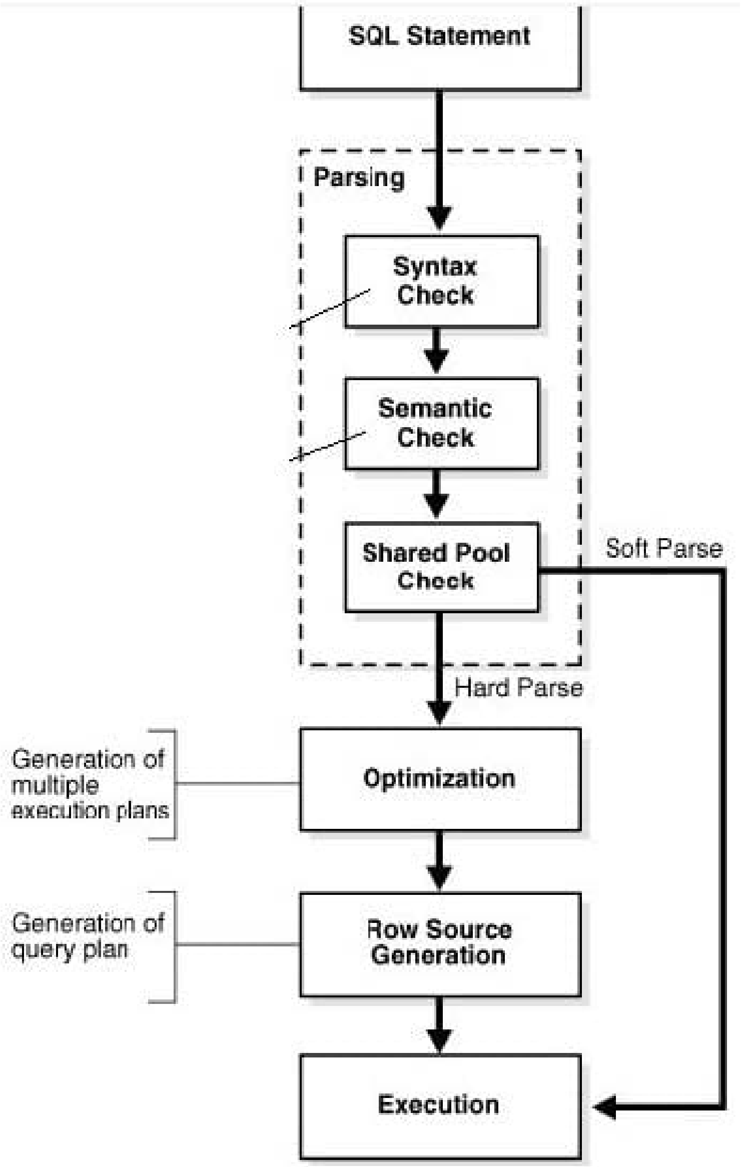
Step 3: Shared pool check

Step 4: Optimization

Step 5: Row Source Optimization (generation of query plan) (3)

Etc.

Note:



Incorrect:

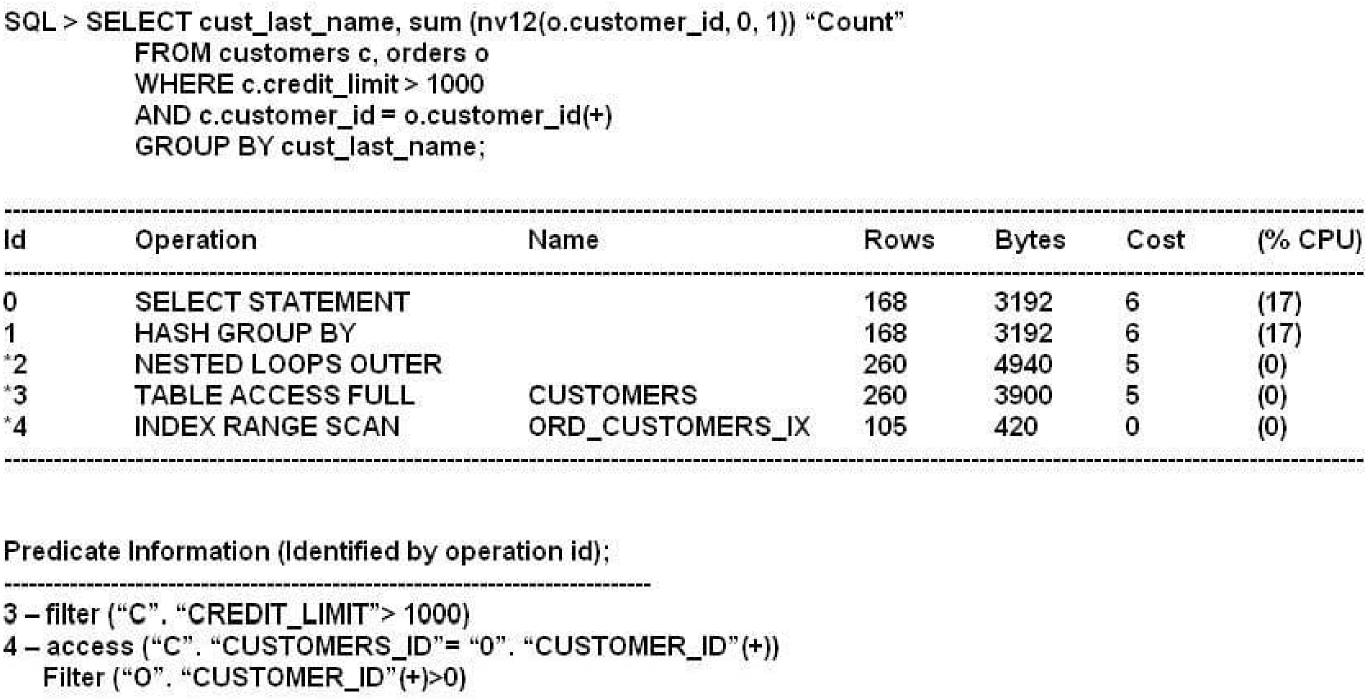
Not A: First execute then fetch rows.

Not B: Check of syntax is before check of semantics.

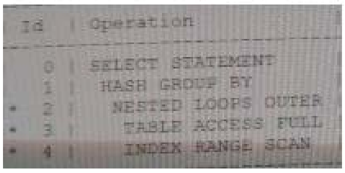
Not D: Parse tree is after semantics.

QUESTION 4

Examine the query and its execution plan: Which two statements are true regarding the execution plan?



```
SQL > SELECT cust_last_name, sum (nv12(o.customer_id, 0, 1)) "Count"
          FROM customers c, orders o
          WHERE c.credit_limit > 1000
          AND c.customer_id = o.customer_id(+)
          GROUP BY cust_last_name;

-----------------------------------------------------------------------------------------------------
Id        Operation                    Name                  Rows     Bytes     Cost     (% CPU)
-----------------------------------------------------------------------------------------------------
0         SELECT STATEMENT                                   168      3192      6        (17)
1         HASH GROUP BY                                      168      3192      6        (17)
*2        NESTED LOOPS OUTER                                 260      4940      5        (0)
*3        TABLE ACCESS FULL            CUSTOMERS             260      3900      5        (0)
*4        INDEX RANGE SCAN             ORD_CUSTOMERS_IX      105      420       0        (0)
-----------------------------------------------------------------------------------------------------

Predicate Information (Identified by operation id);
---------------------------------------------------
3 – filter ("C". "CREDIT_LIMIT"> 1000)
4 – access ("C". "CUSTOMERS_ID"= "O". "CUSTOMER_ID"(+))
   Filter ("O". "CUSTOMER_ID"(+)>0)
```



```
  Id  | Operation
-----------------------------
   0  | SELECT STATEMENT
   1  |  HASH GROUP BY
*  2  |   NESTED LOOPS OUTER
*  3  |    TABLE ACCESS FULL
*  4  |    INDEX RANGE SCAN
```

A. For every row of CUSTOMERS table, the row matching the join predicate from the ORDERS table are returned.

B. An outer join returns NULL for the ORDERS table columns along with the CUSTOMERS table rows when it does not find any corresponding rows in the ORDER table.

C. The data is aggregated from the ORDERS table before joining to CUSTOMERS.

D. The NESTED LOOP OUTER join is performed because the OPTIMZER_MODE parameter is set to ALL_ROWS.

Correct Answer: BD

B: An outer join extends the result of a simple join. An outer join returns all rows that satisfy the join condition and also returns some or all of those rows from one table for which no rows from the other satisfy the join condition.

Note:

*

All_rows attempts to optimize the query to get the very last row as fast as possible. This makes sense in a stored procedure for example where the client does

not regain control until the stored procedure completes. You don\\'t care if you have to wait to get the first row if the last row gets back to you twice as fast. In a

client server/interactive application you may well care about that.

*

The optimizer uses nested loop joins to process an outer join in the following circumstances:

/ It is possible to drive from the outer table to inner table.

/ Data volume is low enough to make the nested loop method efficient.

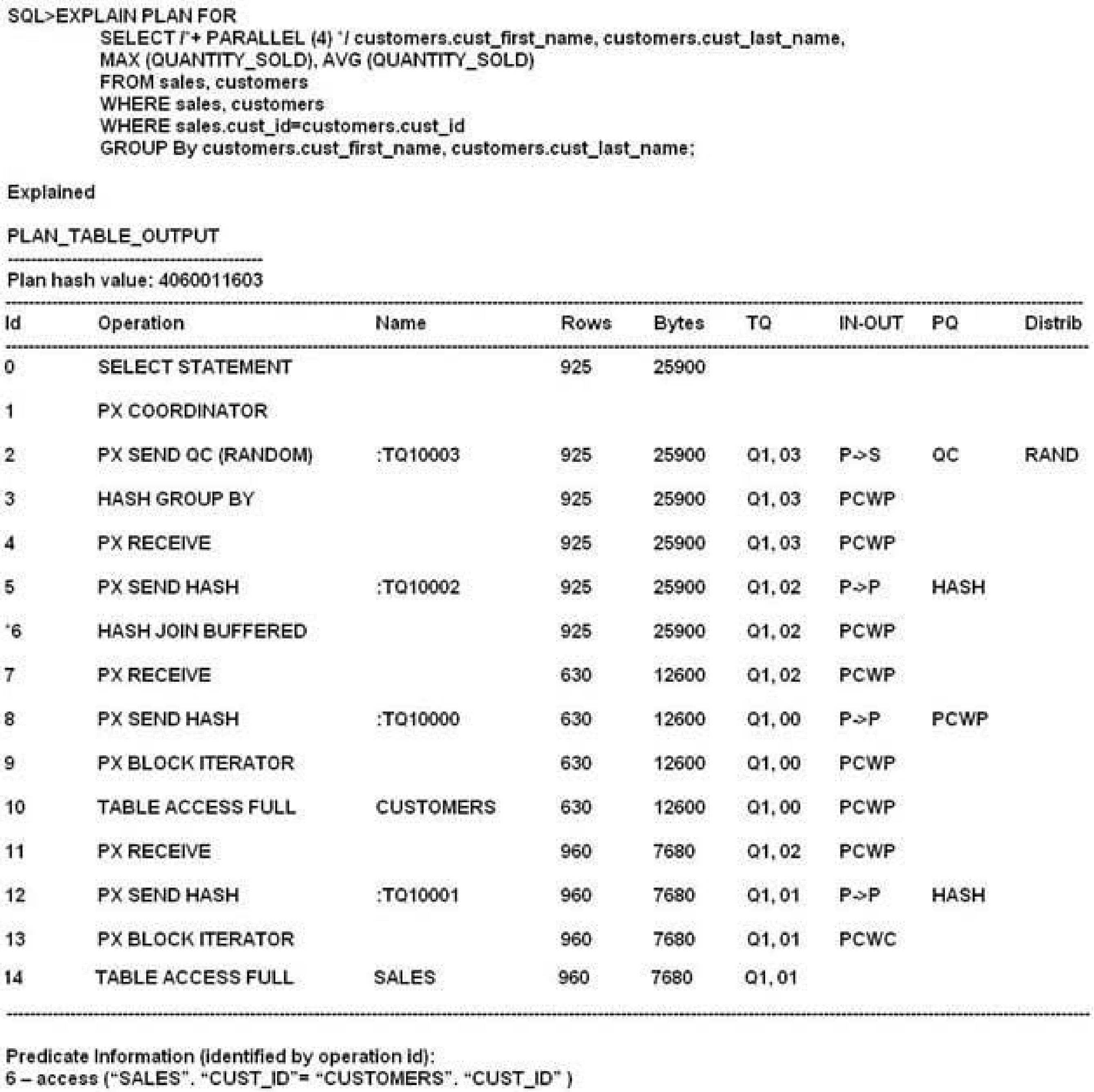
*

First_rows attempts to optimize the query to get the very first row back to the client as fast as possible. This is good for an interactive client server environment

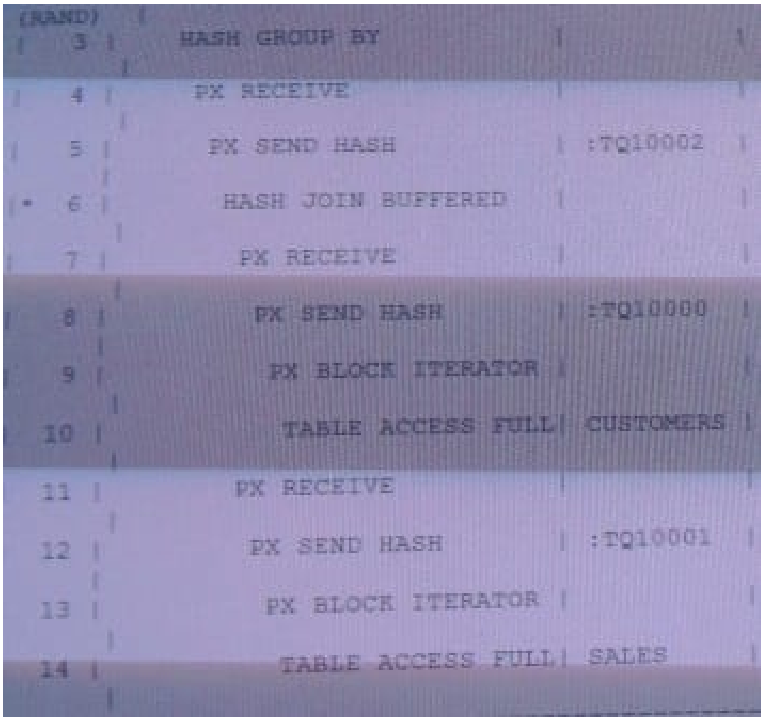
where the client runs a query and shows the user the first 10 rows or so and waits for them to page down to get more.

QUESTION 5

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" )
```



```
(RAND)  |
|   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 us 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 us 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.

Latest 1Z0-117 Dumps   1Z0-117 PDF Dumps   1Z0-117 Study Guide