Free Questions for 1Z0-076 by go4braindumps

Shared by Roman on 29-03-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Which four factors can influence the rate of SQL apply on a logical standby database?

Options:

A- the number of PREPAER processes

B- the number of coordinator processes on the standby database instance

C- the number of full table scans performed by SQL apply

D- the size of the undo tablespace on the logical standby database

E- the number of applier processes

F- the size of the shared pool

Answer:

A, B, C, E

Explanation:

The rate of SQL apply on a logical standby database can be influenced by:

A: The number of PREPARER processes (which seems to be a typographical error and should read as PREPARER or similar) which prepare the redo data for the applier processes.

B: The number of coordinator processes on the standby database instance which coordinate the SQL apply activities.

C: The number of full table scans performed by SQL apply since full table scans can be resource-intensive and slow down the apply rate.

E: The number of applier processes which apply the redo data to the logical standby database.

Option D is incorrect as the size of the undo tablespace on the logical standby database is more likely to affect the SQL apply lag rather than the rate of SQL apply.

Option F is incorrect because the size of the shared pool would typically not influence the rate of SQL apply. The shared pool is more related to the caching of shared SQL and PL/SQL code and control structures.

Question 2

Question Type: **MultipleChoice**

There are currently 6 applief. and 6 pfepafef processes running and no idle applier processes on y logical standby database.

The max_SERVERS SQL apply parameter and number of archiver processes are both set to 12.

Identify two changes, each of which would allow you to increase the number of applier processes.

Options:

A- Decrease the number of archiver processes on the standby database.

B- Increase the processes initialization parameter. D Decrease the number of FREPARER processes.

C- Increase the value for the MAX_SERVERS SQL apply parameter.

D- Increase the parallel_max_server initialization parameter.

E- Increase the RECOVERY_PARALLEL initialization parameter.

Answer:

C, D

Explanation:

To increase the number of applier processes on a logical standby database, the following changes can be made:

C: Increasing the value for the MAX_SERVERS SQL apply parameter would allow for more applier processes to be initiated, assuming that system resources permit.

D: Increasing the PARALLEL_MAX_SERVERS initialization parameter would allow for more parallel execution processes, which can be used by SQL apply to increase the number of applier processes.

Option A is incorrect as decreasing the number of archiver processes will not necessarily increase the number of applier processes; these are unrelated components.

Option B is incorrect because the 'FREPARER' processes do not exist, it seems to be a typographical error, and the 'REPARER' is not a valid Oracle process or parameter.

Option E is incorrect because the RECOVERY_PARALLELISM parameter controls the number of processes used for instance recovery and media recovery, not for SQL apply.

Question 3

Question Type: **MultipleChoice**

You notice that the SQL apply lag on your logical standby database has increased but the redo transport lag has not.

Which four could be reasons for the increase in SQL apply lag?

Options:

A- An undersized undo tablespace on the logical standby

B- Many SQL apply operations do full table scans

C- An increased number of bulk updates on the primary

D- An increased number of bulk inserts on the primary

E- The standby redo log files are undersized on the primary database

F- An undersized shared pool

Answer:

A, B, C, F

Explanation:

The SQL apply lag on a logical standby database can be caused by several factors:

A: An undersized undo tablespace can lead to delays in SQL apply operations as it may not be able to handle the volume of undo records generated by the SQL apply process.

B: SQL apply operations that do full table scans can consume significant system resources, potentially leading to higher apply lag.

C: An increased number of bulk updates on the primary database may generate a large volume of redo data, which can cause apply lag if the logical standby cannot apply the changes quickly enough.

F: An undersized shared pool may affect the parsing and execution of SQL statements by SQL apply, which can contribute to the apply lag.

Option D is less likely to be a direct cause of SQL apply lag compared to bulk updates, as inserts generate new data rather than modifying existing data, which SQL apply can typically handle more efficiently.

Option E is incorrect because the size of the standby redo log files on the primary database impacts redo transport lag, not SQL apply lag.

Question 4

Question Type: MultipleChoice

Which THREE statements are TRUE about Global Sequences when connected to a physical standby database with Real-Time Query enabled?

Options:

A- Their usage will always have a performance impact on the primary database.

B- Their creation requires that a LOG archive_dest_n parameter be defined in the standby that points back to the primary.

C- If the CACHE option is set then the size of the cache must be at least 100.

D- Their usage may have a performance impact on the physical standby database if the CACHE size is too small.

E- They must have the NOORDEK and CACHE options set.

Answer:

A, D, E

Explanation:

Global Sequences are Oracle sequences that generate unique values across multiple instances in an Oracle RAC or a Data Guard configuration. Regarding their behavior and performance when connected to a physical standby database with Real-Time Query enabled:

A: The usage of Global Sequences can indeed have a performance impact on the primary database due to the need to generate unique values that are consistent across both primary and standby databases.

D: The performance impact on the physical standby database may occur if the CACHE size is too small. This is because the standby database will frequently have to access the primary database to replenish the cache, which can increase the load and potentially lead to performance degradation.

E: Global Sequences should have the NOORDER and CACHE options set. The NOORDER option ensures that sequence numbers are provided without guaranteeing sequence order, thus improving scalability and performance. The CACHE option is used to specify how many sequence values will be held in memory for faster access.

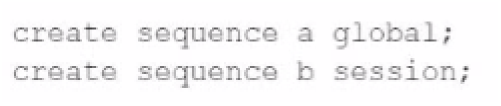
Option B is incorrect as the LOG_ARCHIVE_DEST_n parameter's definition for standbys pointing back to the primary does not directly pertain to the creation of sequences.

Option C is incorrect because there is no requirement that the size of the cache for a sequence must be at least 100. The CACHE size can be set to a different number based on specific use cases or performance considerations.

Question 5

Question Type: **MultipleChoice**

Your Data Guard environment has one physical standby database using Real-Time Query. Two sequences have been created by these SQL statements:



```
create sequence a global;
create sequence b session;
```

Neither sequence has been used since being created.

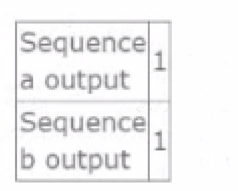
Session 1 connects to the primary database instance and issues these two SQL statements:

SELECT a.nextval FROM DUAL; SELECT b.nextval FROM DUAL;

Then session 2 connects to the physical standby database instance and issues the same SQL statements. Which output will be seen for session 2?

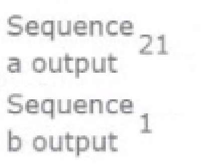
Then session 2 connects to the physical standby database instance and issues the same SQL statements. Which output will be seen for session 2?

A)



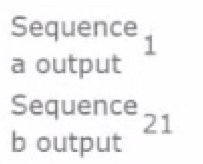
Sequence a output	1
Sequence b output	1

B)

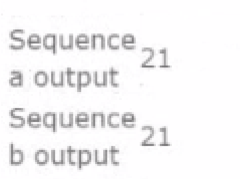


Sequence a output	21
Sequence b output	1

C)



Sequence a output	1
Sequence b output	21



Sequence a output	21
Sequence b output	21

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

C

Explanation:

In Oracle, a sequence created with the GLOBAL keyword is available and can produce values across all sessions and instances. However, a sequence created with the SESSION keyword is only specific to the session it was created in. When the NEXTVAL is called for a sequence, it will increment according to the sequence's properties set during its creation.

Given the sequence creation statements and the actions performed:

The a sequence is global, which means it is available across the entire database, including the standby database with Real-Time Query enabled. So, when session 2 calls a.nextval, it will get the next value in the sequence, which is 21 since session 1 already retrieved 1.

The b sequence is session-specific, so when session 2 calls b.nextval, it will get the value 1 because for this new session on the standby, this is the first time the sequence is being accessed.

Therefore, the output for session 2 will be a output as 21 and b output as 1, which corresponds to Option C.

Question 6

Question Type: **MultipleChoice**

Which THREE statements are TRUE about the supported workload in Active Data Guard standby databases?

Options:

A- PL/SQL blocks that you run on Active Data Guard standby databases can be always redirected to and run on the primary database.

B- Read-mostly reporting applications that use global temporary tables for storing temporary data can be offloaded.

C- You might have to use sequences with global temporary tables to support read-mostly applications by using Active Data Guard.

D- The DDL operations on private temporary tables are transparently redirected to the primary database.

E- The DML operations on a standby can be transparently redirected to and run on the primary database

Answer:

B, C, E

Explanation:

In an Oracle Active Data Guard environment:

B: Read-mostly reporting applications that utilize global temporary tables to store session-specific data can be effectively offloaded to an Active Data Guard standby database, reducing the load on the primary database.

C: Sequences can be used with global temporary tables on an Active Data Guard standby database to support certain types of read-mostly applications, though some restrictions on sequence use may apply.

E: In Oracle Database 19c and later, DML redirection allows DML operations performed on an Active Data Guard standby database to be transparently redirected to the primary database. This is part of the DML Redirection feature.

Option A is incorrect because not all PL/SQL blocks run on an Active Data Guard standby database can be redirected to the primary database. Some PL/SQL executions, specifically those that would attempt to make changes to the database, are not supported on the standby.

Option D is incorrect because DDL operations on private temporary tables are not redirected; instead, private temporary tables are session-specific and are not persisted on disk, so they do not generate redo and are not applicable to an Active Data Guard standby.