Free Questions for ADA-C01 by vceexamstest

Shared by Kelly on 29-01-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

A company enabled replication between accounts and is ready to replicate data across regions in the same cloud service provider.

The primary database object is : PROD_AWS_EAST. Location : AWS_EAST

The secondary database object is : PROD_AWS_WEST. Location : AWS_WEST

What command and account location is needed to refresh the data?

Options:

A- Location : AWS_WEST
Command : REFRESH DATABASE PROD_AWS WEST REFRESH;

B- Location : AWS_WEST
Command : ALTER DATABASE PROD AWS WEST REFRESH;

C- Location : AWS_EAST
Command : REFRESH DATABASE PROD_AWS_WEST REFRESH;

D- Location : AWS EAST
Command: ALTER DATABASE PROD_AWS_WEST REFRESH;

Answer:

A

Explanation:

The REFRESH DATABASE command is used to refresh a secondary database with the latest data and metadata from the primary database1. The command must be executed in the target account where the secondary database resides2. Therefore, the answer is A, as the location is AWS_WEST and the command is REFRESH DATABASE PROD_AWS_WEST REFRESH. The other options are incorrect because they either use the wrong location, the wrong command, or the wrong database name.

Question 2

Question Type: MultipleChoice

A user has enrolled in Multi-factor Authentication (MFA) for connecting to Snowflake. The user informs the Snowflake Administrator that they lost their mobile phone the previous evening.

Which step should the Administrator take to allow the user to log in to the system, without revoking their MFA enrollment?

Options:

A- Alter the user and set MINS TO BYPASS MFA to a value that will disable MFA long enough for the user to log in.

B- Alter the user and set DISABLE_MFA to true, which will suspend the MFA requirement for 24 hours.

C- Instruct the user to connect to Snowflake using SnowSQL, which does not support MFA authentication.

D- Instruct the user to append the normal URL with /?mode=mfa_bypass&code= to log on.

Answer:

A

Explanation:

The MINS_TO_BYPASS_MFA property allows the account administrator to temporarily disable MFA for a user who has lost their phone or changed their phone number1. The user can log in without MFA for the specified number of minutes, and then re-enroll in MFA using their new phone1. This does not revoke their MFA enrollment, unlike the DISABLE_MFA property, which cancels their enrollment and requires them to re-enroll from scratch1. The other options are not valid ways to bypass MFA, as SnowSQL does support MFA authentication2, and there is no such URL parameter as /?mode=mfa_bypass&code= for Snowflake3

Question 3

Question Type: MultipleChoice

What session parameter can be used to test the integrity of secure views based on the account that is accessing that view?

Options:

A- MIMIC_CONSUMER_ACCOUNT

B- TEST_ACCOUNT_ID

C- PRODUCER_TEST_ACCT

D- SIMULATED_DATA_SHARING_CONSUMER

Answer:

D

Explanation:

The SIMULATED_DATA_SHARING_CONSUMER session parameter allows a data provider to test the integrity of secure views based on the account that is accessing that view2. By setting this parameter to the name of the consumer account, the data provider can query the secure view and see the results that a user in the consumer account will see2. This helps to ensure that sensitive data in a shared database is not exposed to unauthorized users1. The other options are not valid session parameters in Snowflake3

Question 4

Question Type: MultipleChoice

Which masking policy will mask a column whenever it is queried through a view owned by a role named MASKED_VIEW_ROLE?

Options:

A- create or replace masking policy maskstring as (val string) returns string ->
case
when is_role_in_session ('MASKED_VIEW_ROLE') then ' **
else val
end;
*,

B- create or replace masking policy maskString as (val string) returns string ->
case
when array_contains ('MASKED_VIEW_ROLE' :: variant, parse_json (current_available_roles ())) then '*
else val
end;
** '

C- create or replace masking policy maskstring as (val string) returns string ->
case

when invoker_role() in ('MASKED_VIEW_ROLE') then
else val
end;
' **

D- create or replace masking policy maskString as (val string) returns string ->
case
when current_role() in ('MASKED_VIEW_ROLE') then ' ********* '
else val
end;

Answer:

A

Explanation:

A masking policy is a SQL expression that transforms the data in a column based on the role that queries the column1. The is_role_in_session function returns true if the specified role is in the current session2. Therefore, the masking policy in option A will mask the column data with asterisks whenever it is queried through a view owned by the MASKED_VIEW_ROLE3. The other options use different functions that do not check the ownership of the view, but rather the current role, the invoker role, or the available roles in the session45. These functions may not return the desired result if the role that owns the view is different from the role that queries the view.

Question 5

Question Type: MultipleChoice

Which Snowflake objects can be managed using SCIM integration? (Select TWO).

Options:

A- Stages

B- Users

C- Warehouses

D- Roles

E- Shares

Answer:

B, D

Explanation:

A SCIM security integration allows the automated management of user identities and groups (i.e. roles) by creating an interface between Snowflake and a third-party Identity Provider (IdP)1. Snowflake supports SCIM integration with Okta, Azure, and custom SCIM clients2.

SCIM integration does not support managing other Snowflake objects, such as stages, warehouses, or shares3. Therefore, the answer is B. Users and D. Roles.

Question 6

Question Type: MultipleChoice

If the query matches the definition, will Snowflake always dynamically rewrite the query to use a materialized view?

Options:

A- No, because joins are not supported by materialized views.

B- No, because the optimizer might decide against it.

C- No, because the materialized view may not be up-to-date.

D- Yes, because materialized views are always faster.

Answer:

B

Explanation:

Snowflake's query optimizer can automatically rewrite queries against the base table or regular views to use the materialized view instead, if the query matches the definition of the materialized view1. However, this is not always guaranteed, as the optimizer might decide against using the materialized view based on various factors, such as the freshness of the data, the size of the result set, the complexity of the query, and the availability of the materialized view2. Therefore, the answer is no, because the optimizer might decide against it.

Question 7

Question Type: MultipleChoice

An Administrator is evaluating a complex query using the EXPLAIN command. The Globalstats operation indicates 500 partitionsAssigned.

The Administrator then runs the query to completion and opens the Query Profile. They notice that the partitions scanned value is 429.

Why might the actual partitions scanned be lower than the estimate from the EXPLAIN output?

Options:

A- The EXPLAIN results always include a 10-15% safety factor in order to provide conservative estimates.

B- The GlobalStats partition assignment includes the micro-partitions that will be assigned for preservation of the query results.

C- Runtime optimizations such as join pruning can reduce the number of partitions and bytes scanned during query execution.

D- In-flight data compression will result in fewer micro-partitions being scanned at the virtual warehouse layer than were identified at the storage layer.

Answer:

C

Explanation:

The EXPLAIN command returns the logical execution plan for a query, which shows the upper bound estimates for the number of partitions and bytes that might be scanned by the query1. However, these estimates do not account for the runtime optimizations that Snowflake performs to improve the query performance and reduce the resource consumption2. One of these optimizations is join pruning, which eliminates unnecessary partitions from the join inputs based on the join predicates2. This can result in fewer partitions and bytes scanned than the estimates from the EXPLAIN output3. Therefore, the actual partitions scanned value in the Query Profile can be lower than the partitionsAssigned value in the EXPLAIN output4.

Question 8

Question Type: MultipleChoice

What access control policy will be put into place when future grants are assigned to both database and schema objects?

Options:

A- Database privileges will take precedence over schema privileges.

B- Schema privileges will take precedence over database privileges.

C- An access policy combining both the database object and the schema object will be used, with the most permissive policy taking precedence.

D- An access policy combining both the database object and the schema object will be used, with the most restrictive policy taking precedence.

Answer:

B

Explanation:

When future grants are defined on the same object type for a database and a schema in the same database, the schema-level grants take precedence over the database level grants, and the database level grants are ignored4. This behavior applies to privileges on future objects granted to one role or different roles4. Future grants allow defining an initial set of privileges to grant on new (i.e. future) objects of a certain type in a database or a schema3. As soon as the new objects are created inside the database or schema, the

predefined set of privileges are assigned to the object automatically without any manual intervention3.

Question 9

Question Type: MultipleChoice

An Administrator loads data into a staging table every day. Once loaded, users from several different departments perform transformations on the data and load it into

different production tables.

How should the staging table be created and used to MINIMIZE storage costs and MAXIMIZE performance?

Options:

A- Create it as an external table, which will not incur Time Travel costs.

B- Create it as a transient table with a retention time of 0 days.

C- Create it as a temporary table with a retention time of 0 days.

D- Create it as a permanent table with a retention time of 0 days.

Answer:

B

Explanation:

According to the Snowflake documentation1, a transient table is a type of table that does not support Time Travel or Fail-safe, which means that it does not incur any storage costs for maintaining historical versions of the data or backups for disaster recovery. A transient table can be dropped at any time, and the data is not recoverable. A transient table can also have a retention time of 0 days, which means that the data is deleted immediately after the table is dropped or truncated. Therefore, creating the staging table as a transient table with a retention time of 0 days can minimize the storage costs and maximize the performance, as the data is only loaded and transformed once, and then deleted after the production tables are populated. Option A is incorrect because creating the staging table as an external table, which references data files stored in a cloud storage location, can incur additional costs and complexity for data transfer and synchronization, and may not provide the best performance for data loading and transformation. Option C is incorrect because creating the staging table as a temporary table, which is automatically dropped when the session ends or the user logs out, can cause data loss or inconsistency if the session is interrupted or terminated before the production tables are populated. Option D is incorrect because creating the staging table as a permanent table, which supports Time Travel and Fail-safe, can incur additional storage costs for maintaining historical versions of the data and backups for disaster recovery, and may not provide the best performance for data loading and transformation.

Question 10

Question Type: MultipleChoice

A user with the proper role issues the following commands when setting up and activating network policies:

CREATE OR REPLACE NETWORK POLICY foo_policy

ALLOWED_IP_LIST = ('1.1.1.0/24', '2.2.2.0/24' , '3.3. 3. 0/24')

BLOCKED IP LIST = ('1.1.1.1')

COMMENT = 'Account level policy';

ALTER ACCOUNT SET NETWORK_POLICY=FOO_POLICY;

CREATE OR REPLACE NETWORK POLICY bar_policy

ALLOWED_IP_LIST = ('3.3.3.0/24')

BLOCKED IP LIST = ('3.3.3.10')

COMMENT = 'user level policy';

ALTER USER userl SET NETWORK_POLICY=BAR_POLICY;

Afterwards, user1 attempts to log in to Snowflake from IP address 3.3.3.10.

Will the login be successful?

Options:

A- Yes, because 3.3.3.10 is found in the ALLOWED_IP_LIST of bar_policy.

B- No, because 3.3.3.10 is found in the BLOCKED_IP_LIST of bar_policy.

C- Yes, because 3.3.3.10 is found in the ALLOWED_IP_LIST of foo_policy.

D- No, because 3.3.3.10 is not found in the ALLOWED_IP_LIST of foo_policy.

Answer:

B

Explanation:

According to the Snowflake documentation1, network policies are a feature that allows restricting access to your account based on user IP address. A network policy can be applied to an account, a user, or a security integration, and can specify a list of allowed IP addresses and a list of blocked IP addresses. If there are network policies applied to more than one of these, the most specific network policy overrides more general network policies. In this case, the user1 has a network policy (bar_policy) applied to them, which overrides the account-level network policy (foo_policy). The bar_policy allows access only from the IP range 3.3.3.0/24, and blocks access from the IP address 3.3.3.10. Therefore, the user1 will not be able to log in to Snowflake from IP address 3.3.3.10, as it is found in the BLOCKED_IP_LIST of bar_policy. Option A is incorrect because the ALLOWED_IP_LIST of bar_policy does not override the BLOCKED_IP_LIST of bar_policy. Option C is incorrect because the ALLOWED_IP_LIST of foo_policy does not apply to user1, as it is overridden by the user-level network policy. Option D is incorrect because the ALLOWED_IP_LIST of foo_policy does not matter, as it is overridden by the user-level network policy.