ATMIC NETWORKS
YOUR GUIDE TO GLOBAL CERTIFIATION

"Please note that these files may not be up to date. However, the questions will help you understand the exam format and typical question patterns."

www.atmicnetworks .com

Warning: Keep connected with our support team for latest updates

Question: 1

When a role is dropped, which role inherits ownership of objects owned by the dropped role?

A. The SYSADMIN role

B. The role above the dropped role in the RBAC hierarchy

C. The role executing the command

D. The SECURITYADMIN role

Answer: B

Explanation:

According to the Snowflake documentation1, when a role is dropped, ownership of all objects owned by the dropped role is transferred to the role that is directly above the dropped role in the role hierarchy. This is to ensure that there is always a single owner for each object in the system.

1: Drop Role | Snowflake Documentation

Question: 2

Company A uses Snowflake to manage audio files of call recordings. Company A hired Company B, who also uses Snowflake, to transcribe the audio files for further analysis.

Company A's Administrator created a share.

What object should be added to the share to allow Company B access to the files?

A. A secure view with a column for file URLs.

B. A secure view with a column for pre-signed URLs.

C. A secure view with a column for METADATA$FILENAME.

D. A secure view with a column for the stage name and a column for the file path.

Answer: B

Explanation:

According to the Snowflake documentation1, pre-signed URLs are required to access external files in a share. A secure view can be used to generate pre-signed URLs for the audio files stored in an external stage and expose them to the consumer account. Option A is incorrect because file URLs alone are

not sufficient to access external files in a share. Option C is incorrect because METADATA$FILENAME only returns the file name, not the full path or URL. Option D is incorrect because the stage name and file path are not enough to generate pre-signed URLs.

Question: 3

A retailer uses a TRANSACTIONS table (100M rows, 1.2 TB) that has been clustered by the STORE_ID column (varchar(50)). The vast majority of analyses on this table are grouped by STORE_ID to look at store performance.

There are 1000 stores operated by the retailer but most sales come from only 20 stores. The Administrator notes that most queries are currently experiencing poor pruning, with large amounts of bytes processed by even simple queries.

Why is this occurring?

A. The STORE_ID should be numeric.

B. The table is not big enough to take advantage of the clustering key.

C. Sales across stores are not uniformly distributed.

D. The cardinality of the stores to transaction count ratio is too low to use the STORE_ID as a clustering key.

Answer: C

Explanation:

According to the Snowflake documentation1, clustering keys are most effective when the data is evenly distributed across the key values. If the data is skewed, such as in this case where most sales come from only 20 stores out of 1000, then the micro-partitions will not be well-clustered and the pruning will be poor. This means that more bytes will be scanned by queries, even if they filter by STORE_ID. Option A is incorrect because the data type of the clustering key does not affect the pruning. Option B is incorrect because the table is large enough to benefit from clustering, if the data was more balanced. Option D is incorrect because the cardinality of the clustering key is not relevant

for pruning, as long as the key values are distinct.

1: Considerations for Choosing Clustering for a Table | Snowflake Documentation

Question: 4

A team is provisioning new lower environments from the production database using cloning. All production objects and references reside in the database, and do not have external references.

What set of object references needs to be re-pointed before granting access for usage?

A. Sequences, views, and secure views

B. Sequences, views, secure views, and materialized views

C. Sequences, storage integrations, views, secure views, and materialized views

D. There are no object references that need to be re-pointed

Answer: C

Explanation:

According to the Snowflake documentation1, when an object in a schema is cloned, any future grants defined for this object type in the schema are applied to the cloned object unless the COPY GRANTS option is specified in the CREATE statement for the clone operation. However, some objects may still reference the source object or external objects after cloning, which may cause issues with access or functionality. These objects include:

- Sequences: If a table column references a sequence that generates default values, the cloned table may reference the source or cloned sequence, depending on where the sequence is defined. To avoid conflicts, the sequence reference should be re-pointed to the desired sequence using the ALTER TABLE command2.
- Storage integrations: If a stage or a table references a storage integration, the cloned object may still reference the source storage integration, which may not be accessible or valid in the new environment. To avoid errors, the storage integration reference should be re-pointed to the desired storage integration using the ALTER STAGE or ALTER TABLE command34.
- Views, secure views, and materialized views: If a view references another view or table, the cloned view may still reference the source object, which may not be accessible or valid in the new environment. To avoid errors, the view reference should be re-pointed to the desired object using the CREATE OR REPLACE VIEW command5.
- : Cloning Considerations | Snowflake Documentation 2: [ALTER TABLE | Snowflake Documentation] 3: [ALTER STAGE | Snowflake Documentation] 4: [ALTER TABLE | Snowflake Documentation] 5: [CREATE VIEW | Snowflake Documentation]

Question: 5

Which function is the role SECURITYADMIN responsible for that is not granted to role USERADMIN?

A. Reset a Snowflake user's password

B. Manage system grants

C. Create new users

D. Create new roles

Answer: B

Explanation:

According to the Snowflake documentation1, the SECURITYADMIN role is responsible for managing all grants on objects in the account, including system grants. The USERADMIN role can only create and manage users and roles, but not grant privileges on other objects. Therefore, the function that is unique to the SECURITYADMIN role is to manage system grants. Option A is incorrect because both roles can reset a user's password. Option C is incorrect because both roles can create new users. Option D is incorrect because both roles can create new roles.

Question: 6

An Administrator has a table named SALES_DATA which needs some edits, but the Administrator does not want to change the main table data. The Administrator decides to make a transient copy of this table and wants the transient table to have all the same permissions as the original table.

How can the Administrator create the transient table so it inherits the same permissions as the original table, and what considerations need to be made concerning the requirements? (Select TWO).

A. Use the following SQL command:
create transient table TRANSIENT_SALES_DATA as select * from SALES_DATA;

B. Use the following SQL command:
create transient table TRANSIENT SALES DATA as select * from SALES_DATA copy grants;

C. Use the following SQL commands:
create transient table TRANSIENT_SALES_DATA like SALES_DATA copy grants;
insert into TRANSIENT_SALES_DATA select * from SALES_DATA;

D. Transient tables will persist until explicitly dropped and contribute to overall storage costs.

E. Transient tables will be purged at the end of the user session and do not have any Fail-safe period.

Answer: BD

Explanation:

According to the Snowflake documentation1, the COPY GRANTS option can be used to copy all privileges, except OWNERSHIP, from the existing table to the new transient table. This option also preserves any future grants defined for the object type in the schema. Option A is incorrect because it does not copy any grants from the original table. Option C is incorrect because it does not copy the data from the original table, only the structure and grants. Option E is incorrect because transient tables are not session-based and do not have a Fail-safe period, but they do have a Time Travel retention period2.
1: CREATE TABLE | Snowflake Documentation 2: Working with Temporary and Transient Tables | Snowflake Documentation

Question: 7

Which actions are considered breaking changes to data that is shared with consumers in the Snowflake Marketplace? (Select TWO).

A. Dropping a column from a table

B. Deleting data from a table

C. Unpublishing the data listing

D. Renaming a table

E. Adding region availability to the listing

Answer: AD

Explanation:

According to the Snowflake documentation1, breaking changes are changes that affect the schema or structure of the shared data, such as dropping or renaming a column or a table. These changes may cause errors or unexpected results for the consumers who query the shared data. Deleting data from a table, unpublishing the data listing, or adding region availability to the listing are not breaking changes, as they do not alter the schema or structure of the shared data.

1: Managing Data Listings in Snowflake Data Marketplace | Snowflake Documentation

Question: 8

What are the MINIMUM grants required on the database, schema, and table for a stream to be properly created and managed?

A. Database: Usage
Schema: Usage
Table: Select, Create Stream

B. Database: Usage
Schema: Usage
Table: Select

C. Database: Usage, Create Stream
Schema: Usage
Table: Select

D. Database: Usage
Schema: Usage, Create Stream
Table: Select

Answer: A

Explanation:

Question: 9

An Administrator has been asked to support the company's application team need to build a loyalty program for its customers. The customer table contains Personal Identifiable Information (PII), and the application team's role is DEVELOPER.

CREATE TABLE customer_data (customer_first_name string, customer_last_name string, customer_address string, customer_email string, ... some other columns,);

The application team would like to access the customer data, but the email field must be obfuscated.

How can the Administrator protect the sensitive information, while maintaining the usability of the data?

A. Create a view on the customer_data table to eliminate the email column by omitting it from the SELECT clause. Grant the role DEVELOPER access to the view.

B. Create a separate table for all the non-Pll columns and grant the role DEVELOPER access to the new table.

C. Use the CURRENT_ROLE and CURRENT_USER context functions to integrate with a secure view

and filter the sensitive data.

D. Use the CURRENT_ROLE context function to integrate with a masking policy on the fields that contain sensitive data.

Answer: D

Explanation:

Question: 10

An organization's sales team leverages this Snowflake query a few times a day:

SELECT CUSTOMER ID, CUSTOMER_NAME, ADDRESS, PHONE NO
FROM CUSTOMERS
WHERE LAST UPDATED BETWEEN TO_DATE (CURRENT_TIMESTAMP) AND (TO_DATE (CURRENT_TIMESTAMP) -7);

What can the Snowflake Administrator do to optimize the use of persisted query results whenever possible?

A. Wrap the query in a User-Defined Function (UDF) to match syntax execution.

B. Assign everyone on the sales team to the same virtual warehouse.

C. Assign everyone on the sales team to the same security role.

D. Leverage the CURRENT_DATE function for date calculations.

Answer: D

Explanation:

According to the web search results from my predefined tool search_web, one of the factors that affects the reuse of persisted query results is the exact match of the query syntax1. If the query contains functions that return different values for successive runs, such as CURRENT_TIMESTAMP, then the query will not match the previous query and will not benefit from the cache. To avoid this, the query should use functions that return consistent values for the same day, such as CURRENT_DATE, which returns the current date without the time component2. Option A is incorrect because wrapping the query in a UDF does not guarantee the syntax match, as the UDF may also contain dynamic functions. Option B is incorrect because the virtual warehouse does not affect the persisted query results, which are stored at the account level1. Option C is incorrect because the security role does not affect the persisted query results, as long as the role has the necessary privileges to access the tables and views used in the query1.

1: Using Persisted Query Results | Snowflake Documentation 2: Date and Time Functions | Snowflake Documentation

Question: 11

Which tasks can be performed by the ORGADMIN role? (Select THREE).

A. Create one or more accounts in the organization.

B. View a list of all regions enabled for the organization.

C. Create secure views on application tables within the organization.

D. View usage information for all accounts in the organization.

E. Perform zero-copy cloning on account data.

F. Create a reader account to share data with another organization.

Answer: ABD

Explanation:

A user with the ORGADMIN role can perform the following tasks1:

- Create one or more accounts in the organization.
- View a list of all regions enabled for the organization.

- View usage information for all accounts in the organization.

Option C is incorrect because creating secure views on application tables is not a function of the ORGADMIN role, but rather a function of the roles that have access to the tables and schemas within the accounts. Option E is incorrect because performing zero-copy cloning on account data is not a function of the ORGADMIN role, but rather a function of the roles that have the CLONE privilege on the objects within the accounts. Option F is incorrect because creating a reader account to share data with another organization is not a function of the ORGADMIN role, but rather a function of the roles that have the CREATE SHARE privilege on the objects within the accounts.

Question: 12

What role or roles should be used to properly create the object required to setup OAuth 2.0 integration?

A. Any role with GRANT USAGE on SECURITY INTEGRATION

B. ACCOUNTADMIN and SYSADMIN

C. ACCOUNTADMIN and SECURITYADMIN

D. ACCOUNTADMIN only

Answer: D

Explanation:

According to the Using OAuth 2.0 with Snowflake - Blog, only the ACCOUNTADMIN role can create and manage integrations, so an administrator must assume that role when creating a security integration for OAuth. The other roles do not have the necessary privileges to create the object required to setup OAuth 2.0 integration.

Question: 13

The following SQL command was executed:

Use role SECURITYADMIN;

Grant ownership
On future tables
In schema PROD. WORKING
To role PROD_WORKING_OWNER;

Grant role PROD_WORKING_OWNER to role SYSADMIN;

Use role ACCOUNTADMIN;

Create table PROD.WORKING.XYZ (value number) ;

Which role(s) can alter or drop table XYZ?

A. Because ACCOUNTADMIN created the table, only the ACCOUNTADMIN role can alter or drop table XYZ.

B. SECURITYADMIN, SYSADMIN, and ACCOUNTADMIN can alter or drop table XYZ.

C. PROD_WORKING_OWNER, ACCOUNTADMIN, and SYSADMIN can alter or drop table XYZ.

D. Only the PROD_WORKING_OWNER role can alter or drop table XYZ.

Answer: C

Explanation:

According to the GRANT OWNERSHIP documentation, the ownership privilege grants full control over the table and can only be held by one role at a time. However, the current owner can also grant the ownership privilege to another role, which transfers the ownership to the new role. In this case,

the SECURITYADMIN role granted the ownership privilege on future tables in the PROD.WORKING schema to the PROD_WORKING_OWNER role. This means that any table created in that schema after the grant statement will be owned by the PROD_WORKING_OWNER role. Therefore, the PROD_WORKING_OWNER role can alter or drop table XYZ, which was created by the ACCOUNTADMIN role in the PROD.WORKING schema. Additionally, the ACCOUNTADMIN role can also alter or drop table XYZ, because it is the top-level role that has all privileges on all objects in the account. Furthermore, the SYSADMIN role can also alter or drop table XYZ, because it was granted the PROD_WORKING_OWNER role by the SECURITYADMIN role. The SYSADMIN role can activate the PROD_WORKING_OWNER role and inherit its privileges, including the ownership privilege on table XYZ. The SECURITYADMIN role cannot alter or drop table XYZ, because it does not have the ownership privilege on the table, nor does it have the PROD_WORKING_OWNER role.

Question: 14

When adding secure views to a share in Snowflake, which function is needed to authorize users from another account to access rows in a base table?

A. CURRENT_ROLE

B. CURRENT ACCOUNT

C. CURRENT_USER

D. CURRENT_CLIENT

Answer: C

Explanation:

According to the Working with Secure Views documentation, secure views are designed to limit access to sensitive data that should not be exposed to all users of the underlying table(s). When sharing secure views

with another account, the view definition must include a function that returns the identity of the user who is querying the view, such as CURRENT_USER, CURRENT_ROLE, or CURRENT_ACCOUNT. These functions can be used to filter the rows in the base table based on the user's identity. For example, a secure view can use the CURRENT_USER function to compare the user name with a column in the base table that contains the authorized user names. Only the rows that match the user name will be returned by the view. The CURRENT_CLIENT function is not suitable for this purpose, because it returns the IP address of the client that is connected to Snowflake, which is not related to the user's identity.

Question: 15

In which scenario will use of an external table simplify a data pipeline?

A. When accessing a Snowflake table from a relational database

B. When accessing a Snowflake table from an external database within the same region

C. When continuously writing data from a Snowflake table to external storage

D. When accessing a Snowflake table that references data files located in cloud storage

Answer: D

Explanation:

According to the Introduction to External Tables documentation, an external table is a Snowflake feature that allows you to query data stored in an external stage as if the data were inside a table in Snowflake. The external stage is not part of Snowflake, so Snowflake does not store or manage the stage. This simplifies the data pipeline by eliminating the need to load the data into Snowflake before querying it. External tables can access data stored in any format that the COPY INTO <table> command supports, such as CSV, JSON, AVRO, ORC, or PARQUET. The other scenarios do not involve external tables, but rather require data loading, unloading, or federation.

Question: 16

A Snowflake user runs a complex SQL query on a dedicated virtual warehouse that reads a large amount of data from micro-partitions. The same user wants to run another query that uses the same data set.

Which action would provide optimal performance for the second SQL query?

A. Assign additional clusters to the virtual warehouse.

B. Increase the STATEMENT_TIMEOUT_IN_SECONDS parameter in the session.

C. Prevent the virtual warehouse from suspending between the running of the first and second queries.

D. Use the RESULT_SCAN function to post-process the output of the first query.

Answer: D

Explanation:

According to the Using Persisted Query Results documentation, the RESULT_SCAN function allows you to query the result set of a previous command as if it were a table. This can improve the performance of the second query by avoiding reading the same data from micro-partitions again. The other actions do not provide optimal performance for the second query because:

- Assigning additional clusters to the virtual warehouse does not affect the data access speed, but only the query execution speed. It also increases the cost of the warehouse.
- Increasing the STATEMENT_TIMEOUT_IN_SECONDS parameter in the session does not improve the performance of the query, but only allows it to run longer before timing out. It also increases the risk of resource contention and deadlock.
- Preventing the virtual warehouse from suspending between the running of the first and second queries does not guarantee that the data will be cached in memory, as Snowflake uses a least recently used (LRU) cache eviction policy. It also increases the cost of the warehouse.

https://docs.snowflake.com/en/user-guide/querying-persisted-results

Question: 17

For Snowflake network policies, what will occur when the account_level and user_level network policies are both defined?

A. The account_level policy will override the user_level policy.

B. The user_level policy will override the account_level policy.

C. The user_level network policies will not be supported.

D. A network policy error will be generated with no definitions provided.

Answer: B

Explanation:

According to the Network Policies documentation, a network policy can be applied to an account, a security integration, or a user. If there are network policies applied to more than one of these, the most specific network policy overrides more general network policies. The following summarizes the order of precedence:

- Account: Network policies applied to an account are the most general network policies. They are overridden by network policies applied to a security integration or user.
- Security Integration: Network policies applied to a security integration override network policies applied to the account, but are overridden by a network policy applied to a user.
- User: Network policies applied to a user are the most specific network policies. They override both accounts and security integrations.

Therefore, if both the account_level and user_level network policies are defined, the user_level policy will take effect and the account_level policy will be ignored. The other options are incorrect because:

- The account_level policy will not override the user_level policy, as explained above.
- The user_level network policies will be supported, as they are part of the network policy feature.
- A network policy error will not be generated, as there is no conflict between the account_level and user_level network policies.

Question: 18

MY_TABLE is a table that has not been updated or modified for several days. On 01 January 2021 at 07:01, a user executed a query to update this table. The query ID is '8e5d0ca9-005e-44e6-b858-a8f5b37c5726'. It is now 07:30 on the same day.

Which queries will allow the user to view the historical data that was in the table before this query was executed? (Select THREE).

A. SELECT * FROM my table WITH TIME_TRAVEL (OFFSET => -60*30);

B. SELECT * FROM my_table AT (TIMESTAMP => '2021-01-01 07:00:00' :: timestamp);

C. SELECT * FROM TIME_TRAVEL ('MY_TABLE', 2021-01-01 07:00:00);

D. SELECT * FROM my table PRIOR TO STATEMENT '8e5d0ca9-005e-44e6-b858-a8f5b37c5726';

E. SELECT * FROM my_table AT (OFFSET => -60*30);

F. SELECT * FROM my_table BEFORE (STATEMENT => '8e5d0ca9-005e-44e6-b858-a8f5b37c5726');

Answer: BDF

Explanation:

According to the AT | BEFORE documentation, the AT or BEFORE clause is used for Snowflake Time Travel, which allows you to query historical data from a table based on a specific point in the past. The clause can use one of the following parameters to pinpoint the exact historical data you wish to access:

- TIMESTAMP: Specifies an exact date and time to use for Time Travel.
- OFFSET: Specifies the difference in seconds from the current time to use for Time Travel.
- STATEMENT: Specifies the query ID of a statement to use as the reference point for Time Travel.

Therefore, the queries that will allow the user to view the historical data that was in the table before the query was executed are:

- B. SELECT * FROM my_table AT (TIMESTAMP => '2021-01-01 07:00:00' :: timestamp); This query uses the TIMESTAMP parameter to specify a point in time that is before the query execution time of 07:01.
- D. SELECT * FROM my table PRIOR TO STATEMENT '8e5d0ca9-005e-44e6-b858-a8f5b37c5726'; This query uses the PRIOR TO STATEMENT keyword and the STATEMENT parameter to specify

a point in time that is immediately preceding the query execution time of 07:01.

- F. SELECT * FROM my_table BEFORE (STATEMENT => '8e5d0ca9-005e-44e6-b858-a8f5b37c5726'); This query uses the BEFORE keyword and the STATEMENT parameter to specify a point in time that is immediately preceding the query execution time of 07:01.

The other queries are incorrect because:

- A. SELECT * FROM my table WITH TIME_TRAVEL (OFFSET => -60*30); This query uses the OFFSET parameter to specify a point in time that is 30 minutes before the current time, which is 07:30. This is after the query execution time of 07:01, so it will not show the historical data before the query was executed.
- C. SELECT * FROM TIME_TRAVEL ('MY_TABLE', 2021-01-01 07:00:00); This query is not valid syntax for Time Travel. The TIME_TRAVEL function does not exist in Snowflake. The correct syntax is to use the AT or BEFORE clause after the table name in the FROM clause.
- E. SELECT * FROM my_table AT (OFFSET => -60*30); This query uses the AT keyword and the OFFSET parameter to specify a point in time that is 30 minutes before the current time, which is 07:30. This is equal to the query execution time of 07:01, so it will not show the historical data before the query was executed. The AT keyword specifies that the request is inclusive of any changes made by a statement or transaction with timestamp equal to the specified parameter. To exclude the changes made by the query, the BEFORE keyword should be used instead.

Question: 19

What are characteristics of Dynamic Data Masking? (Select TWO).

A. A masking policy that is currently set on a table can be dropped.

B. A single masking policy can be applied to columns in different tables.

C. A masking policy can be applied to the VALUE column of an external table.

D. The role that creates the masking policy will always see unmasked data in query results.

E. A single masking policy can be applied to columns with different data types.

Answer: BE

Explanation:

According to the Using Dynamic Data Masking documentation, Dynamic Data Masking is a feature that allows you to alter sections of data in table and view columns at query time using a predefined masking strategy. The following are some of the characteristics of Dynamic Data Masking:

- A single masking policy can be applied to columns in different tables. This means that you can write a policy once and have it apply to thousands of columns across databases and schemas.
- A single masking policy can be applied to columns with different data types. This means that you can use the same masking strategy for columns that store different kinds of data, such as strings, numbers, dates, etc.
- A masking policy that is currently set on a table can be dropped. This means that you can remove the masking policy from the table and restore the original data visibility.

- A masking policy can be applied to the VALUE column of an external table. This means that you can mask data that is stored in an external stage and queried through an external table.
- The role that creates the masking policy will always see unmasked data in query results. This is not true, as the masking policy can also apply to the creator role depending on the execution context conditions defined in the policy. For example, if the policy specifies that only users with a certain custom entitlement can see the unmasked data, then the creator role will also need to have that entitlement to see the unmasked data.

Question: 20

A Snowflake Administrator needs to set up Time Travel for a presentation area that includes facts and dimensions tables, and receives a lot of meaningless and erroneous IoT data. Time Travel is being used as a component of the company's data quality process in which the ingestion pipeline should revert to a known quality data state if any anomalies are detected in the latest load. Data from the past 30 days may have to be retrieved because of latencies in the data acquisition process.

According to best practices, how should these requirements be met? (Select TWO).

A. Related data should not be placed together in the same schema. Facts and dimension tables should each have their own schemas.

B. The fact and dimension tables should have the same DATA_RETENTION_TIME_IN_ DAYS.

C. The DATA_RETENTION_TIME_IN_DAYS should be kept at the account level and never used for lower level containers (databases and schemas).

D. Only TRANSIENT tables should be used to ensure referential integrity between the fact and dimension tables.

E. The fact and dimension tables should be cloned together using the same Time Travel options to reduce potential referential integrity issues with the restored data.

Answer: BE

Explanation:
According to the Understanding & Using Time Travel documentation, Time Travel is a feature that allows you to query, clone, and restore historical data in tables, schemas, and databases for up to 90 days. To meet the requirements of the scenario, the following best practices should be followed: • The fact and dimension tables should have the same DATA_RETENTION_TIME_IN_DAYS. This parameter specifies the number of days for which the historical data is preserved and can be accessed by Time Travel. To ensure that the fact and dimension tables can be reverted to a consistent state in case of any anomalies in the latest load, they should have the same retention period. Otherwise, some tables may lose their historical data before others, resulting in data inconsistency and quality issues.

- The fact and dimension tables should be cloned together using the same Time Travel options to reduce potential referential integrity issues with the restored data. Cloning is a way of creating a copy of an object (table, schema, or database) at a specific point in time using Time Travel. To ensure that the fact and dimension tables are cloned with the same data set, they should be cloned together using the same AT or BEFORE clause. This will avoid any referential integrity issues that may arise from cloning tables at

different points in time.

The other options are incorrect because:

- Related data should not be placed together in the same schema. Facts and dimension tables should each have their own schemas. This is not a best practice for Time Travel, as it does not affect the ability to query, clone, or restore historical data. However, it may be a good practice for data modeling and organization, depending on the use case and design principles.
- The DATA_RETENTION_TIME_IN_DAYS should be kept at the account level and never used for lower level containers (databases and schemas). This is not a best practice for Time Travel, as it limits the flexibility and granularity of setting the retention period for different objects. The retention period can be set at the account, database, schema, or table level, and the most specific setting overrides the more general ones. This allows for customizing the retention period based on the data needs and characteristics of each object.
- Only TRANSIENT tables should be used to ensure referential integrity between the fact and dimension tables. This is not a best practice for Time Travel, as it does not affect the referential integrity between the tables. Transient tables are tables that do not have a Fail-safe period, which means that they cannot be recovered by Snowflake after the retention period ends. However, they still support Time Travel within the retention period, and can be queried, cloned, and restored like permanent tables. The choice of table type depends on the data durability and availability requirements, not on the referential integrity.

Question: 21

A Snowflake Administrator needs to persist all virtual warehouse configurations for auditing and backups. Given a table already exists with the following schema:

Table Name : VWH_META
Column 1 : SNAPSHOT_TIME TIMESTAMP_NTZ
Column 2 : CONFIG VARIANT

Which commands should be executed to persist the warehouse data at the time of execution in JSON format in the table VWH META?

A. 1. SHOW WAREHOUSES;
2. INSERT INTO VWH META
SELECT CURRENT TIMESTAMP (),
FROM TABLE (RESULT_SCAN (LAST_QUERY_ID(1))) ;

B. 1. SHOW WAREHOUSES;
2. INSERT INTO VWH META
SELECT CURRENT TIMESTAMP (), *
FROM TABLE (RESULT_SCAN (LAST_QUERY_ID ())) ;

C. 1. SHOW WAREHOUSES;
2. INSERT INTO VWH_META
SELECT CURRENT_TIMESTAMP (),
OBJECT CONSTRUCT (*)
FROM TABLE (RESULT_SCAN (LAST_QUERY_ID ()));

D. 1. SHOW WAREHOUSES;
2. INSERT INTO VWH META
SELECT CURRENT TIMESTAMP (), *
FROM TABLE (RESULT_SCAN (SELECT
LAST QUERY ID(-1)));

Answer: C

Explanation:

According to the Using Persisted Query Results documentation, the RESULT_SCAN function allows you to query the result set of a previous command as if it were a table. The LAST_QUERY_ID function returns the query ID of the most recent statement executed in the current session. Therefore, the combination of these two functions can be used to access the output of the SHOW WAREHOUSES command, which returns the configurations of all the virtual warehouses in the account. However, to persist the warehouse data in JSON format in the table VWH_META, the OBJECT_CONSTRUCT function is needed to convert the output of the SHOW WAREHOUSES command into a VARIANT column. The OBJECT_CONSTRUCT function takes a list of key-value pairs and returns a single JSON object. Therefore, the correct commands to execute are:

1. SHOW WAREHOUSES;
2. INSERT INTO VWH_META SELECT CURRENT_TIMESTAMP (), OBJECT_CONSTRUCT (*) FROM TABLE (RESULT_SCAN (LAST_QUERY_ID ()));

The other options are incorrect because:

- A. This option does not use the OBJECT_CONSTRUCT function, so it will not persist the warehouse data in JSON format. Also, it is missing the * symbol in the SELECT clause, so it will not select any columns from the result set of the SHOW WAREHOUSES command.
- B. This option does not use the OBJECT_CONSTRUCT function, so it will not persist the warehouse data in JSON format. It will also try to insert multiple columns into a single VARIANT column, which will cause a type mismatch error.
- D. This option does not use the OBJECT_CONSTRUCT function, so it will not persist the warehouse data in JSON format. It will also try to use the RESULT_SCAN function on a subquery, which is not supported. The RESULT_SCAN function can only be used on a query ID or a table name.

Question: 22

What are the requirements when creating a new account within an organization in Snowflake? (Select TWO).

A. The account requires at least one ORGADMIN role within one of the organization's accounts.

B. The account name is immutable and cannot be changed.

C. The account name must be specified when the account is created.

D. The account name must be unique among all Snowflake customers.

E. The account name must be unique within the organization.

Answer: CE

Explanation:

According to the CREATE ACCOUNT documentation, the account name must be specified when the account is created, and it must be unique within an organization, regardless of which Snowflake Region the account is in. The other options are incorrect because:

- The account does not require at least one ORGADMIN role within one of the organization's accounts. The account can be created by an organization administrator (i.e. a user with the ORGADMIN role) through the web interface or using SQL, but the new account does not inherit the ORGADMIN role from the existing account. The new account will have its own set of users, roles, databases, and warehouses.
- The account name is not immutable and can be changed. The account name can be modified by contacting Snowflake Support and requesting a name change. However, changing the account name may affect some features that depend on the account name, such as SSO or SCIM.
- The account name does not need to be unique among all Snowflake customers. The account name only needs to be unique within the organization, as the account URL also includes the region and cloud platform information. For example, two accounts with the same name can exist in different regions or cloud platforms, such as myaccount.us-east-1.snowflakecomputing.com and myaccount.eu-west-1.aws.snowflakecomputing.com.

Question: 23

A Snowflake customer is experiencing higher costs than anticipated while migrating their data warehouse workloads from on-premises to Snowflake. The migration
workloads have been deployed on a single warehouse and are characterized by a large number of small INSERTs rather than bulk loading of large extracts. That single
warehouse has been configured as a single cluster, 2XL because there are many parallel INSERTs that are scheduled during nightly loads.

How can the Administrator reduce the costs, while minimizing the overall load times, for migrating data warehouse history?

A. There should be another 2XL warehouse deployed to handle a portion of the load queries.

B. The 2XL warehouse should be changed to 4XL to increase the number of threads available for parallel load queries.

C. The warehouse should be kept as a SMALL or XSMALL and configured as a multi-cluster warehouse to handle the parallel load queries.

D. The INSERTS should be converted to several tables to avoid contention on large tables that slows down query processing.

Answer: C

Explanation:

According to the Snowflake Warehouse Cost Optimization blog post, one of the strategies to reduce the cost of running a warehouse is to use a multi-cluster warehouse with auto-scaling enabled. This allows the warehouse to automatically adjust the number of clusters based on the concurrency demand and the queue size. A multi-cluster warehouse can also be configured with a minimum and maximum number of clusters, as well as a scaling policy to control the scaling behavior. This way, the warehouse can handle the parallel load queries efficiently without wasting resources or credits. The

blog post also suggests using a smaller warehouse size, such as SMALL or XSMALL, for loading data, as it can perform better than a larger warehouse size for small INSERTs. Therefore, the best option to reduce the costs while minimizing the overall load times for migrating data warehouse history is to keep the warehouse as a SMALL or XSMALL and configure it as a multi-cluster warehouse to handle the parallel load queries. The other options are incorrect because:

- A. Deploying another 2XL warehouse to handle a portion of the load queries will not reduce the costs, but increase them. It will also introduce complexity and potential inconsistency in managing the data loading process across multiple warehouses.
- B. Changing the 2XL warehouse to 4XL will not reduce the costs, but increase them. It will also provide more compute resources than needed for small INSERTs, which are not CPU-intensive but I/O-intensive.
- D. Converting the INSERTs to several tables will not reduce the costs, but increase them. It will also create unnecessary data duplication and fragmentation, which will affect the query performance and data quality.

Question: 24

What roles or security privileges will allow a consumer account to request and get data from the Data Exchange? (Select TWO).

A. SYSADMIN

B. SECURITYADMIN

C. ACCOUNTADMIN

D. IMPORT SHARE and CREATE DATABASE

E. IMPORT PRIVILEGES and SHARED DATABASE

Answer: CD

Explanation:

According to the Accessing a Data Exchange documentation, a consumer account can request and get data from the Data Exchange using either the ACCOUNTADMIN role or a role with the IMPORT SHARE and CREATE DATABASE privileges. The ACCOUNTADMIN role is the top-level role that has all privileges on all objects in the account, including the ability to request and get data from the Data Exchange. A role with the IMPORT SHARE and CREATE DATABASE privileges can also request and get data from the Data Exchange, as these are the minimum privileges required to create a database from a share. The other options are incorrect because:

- A. The SYSADMIN role does not have the privilege to request and get data from the Data Exchange, unless it is also granted the IMPORT SHARE and CREATE DATABASE privileges. The SYSADMIN role is a pre-defined role that has all privileges on all objects in the account, except for the privileges reserved for the ACCOUNTADMIN role, such as managing users, roles, and shares.
- B. The SECURITYADMIN role does not have the privilege to request and get data from the Data Exchange, unless it is also granted the IMPORT SHARE and CREATE DATABASE privileges. The SECURITYADMIN role is a pre-defined role that has the privilege to manage security objects in the account, such as network policies, encryption keys, and security integrations, but not data objects, such as databases, schemas, and tables.
- E. The IMPORT PRIVILEGES and SHARED DATABASE are not valid privileges in Snowflake. The correct privilege names are IMPORT SHARE and CREATE DATABASE, as explained above.

Question: 25

An Administrator wants to delegate the administration of a company's data exchange to users who do not have access to the ACCOUNTADMIN role.

How can this requirement be met?

A. Grant imported privileges on data exchange EXCHANGE_NAME to ROLE_NAME;

B. Grant modify on data exchange EXCHANGE_NAME to ROLE_NAME;

C. Grant ownership on data exchange EXCHANGE_NAME to ROLE NAME;

D. Grant usage on data exchange EXCHANGE_NAME to ROLE_NAME;

Answer: B

Explanation:

According to the [GRANT MODIFY] documentation, the MODIFY privilege on a data exchange allows a role to perform administrative tasks on the data exchange, such as inviting members, approving profiles, and reviewing listings. This privilege can be granted by the ACCOUNTADMIN role or a role that already has the MODIFY privilege on the data exchange. Therefore, to delegate the administration of a company's data exchange to users who do not have access to the ACCOUNTADMIN role, the best option is to grant the MODIFY privilege on the data exchange to a role that the users can assume. The other options are incorrect because:

- A. There is no such privilege as IMPORTED PRIVILEGES in Snowflake. The correct privilege name is IMPORT SHARE, which allows a role to create a database from a share. This privilege is not related to the administration of a data exchange, but to the consumption of shared data.
- C. There is no such privilege as OWNERSHIP in Snowflake. The correct privilege name is OWNED BY, which indicates the role that owns an object and has full control over it. However, this privilege cannot be granted or revoked, but only transferred by the current owner to another role using the GRANT OWNERSHIP command. Therefore, this option is not feasible for delegating the administration of a data exchange.
- D. The USAGE privilege on a data exchange allows a role to access the data exchange and

view the available data listings. This privilege does not allow a role to perform administrative tasks on the data exchange, such as inviting members, approving profiles, and reviewing listings. Therefore, this option is not sufficient for delegating the administration of a data exchange.

Question: 26

A company has implemented Snowflake replication between two Snowflake accounts, both of which are running on a Snowflake Enterprise edition. The replication is for the database APP_DB containing only one schema, APP_SCHEMA. The company's Time Travel retention policy is currently set for 30 days for both accounts. An Administrator has been asked to extend the Time Travel retention policy to 60 days on the secondary database only.

How can this requirement be met?

A. Set the data retentionpolicy on the secondary database to 60 days.

B. Set the data retentionpolicy on the schemas in the secondary database to 60 days.

C. Set the data retentionpolicy on the primary database to 30 days and theschemas to 60 days.

D. Set the data retentionpolicy on the primary database to 60 days.

Answer: A

Explanation:

According to the Replication considerations documentation, the Time Travel retention period for a secondary database can be different from the primary database. The retention period can be set at the database, schema, or table level using the DATA_RETENTION_TIME_IN_DAYS parameter.
Therefore, to extend the Time Travel retention policy to 60 days on the secondary database only, the best option is to set the data retention policy on the secondary database to 60 days using the ALTER DATABASE command. The other options are incorrect because:

- B. Setting the data retention policy on the schemas in the secondary database to 60 days will not affect the database-level retention period, which will remain at 30 days. The most specific setting overrides the more general ones, so the schema-level setting will apply to the tables in the schema, but not to the database itself.
- C. Setting the data retention policy on the primary database to 30 days and the schemas to 60 days will not affect the secondary database, which will have its own retention period. The replication process does not copy the retention period settings from the primary to the secondary database, so they can be configured independently.
- D. Setting the data retention policy on the primary database to 60 days will not affect the secondary database, which will have its own retention period. The replication process does not copy the retention period settings from the primary to the secondary database, so they can be configured independently.

Question: 27

When does auto-suspend occur for a multi-cluster virtual warehouse?

A. When there has been no activity on any cluster for the specified period of time.

B. After a specified period of time when an additional cluster has started on the maximum number of clusters specified for a warehouse.

C. When the minimum number of clusters is running and there is no activity for the specified period of time.

D. Auto-suspend does not apply for multi-cluster warehouses.

Answer: C

Explanation:

According to the Multi-cluster Warehouses documentation, auto-suspend is a feature that allows a warehouse to automatically suspend itself after a specified period of inactivity. For a multi-cluster warehouse, auto-suspend applies to the entire warehouse, not to individual clusters. Therefore, auto-suspend occurs when the minimum number of clusters is running and there is no activity for the specified period of time. The other options are incorrect because:

- A. Auto-suspend does not occur when there has been no activity on any cluster for the specified period of time. This would imply that each cluster has its own auto-suspend timer, which is not the case. The warehouse has a single auto-suspend timer that is reset by any activity on any cluster.
- B. Auto-suspend does not occur after a specified period of time when an additional cluster has started on the maximum number of clusters specified for a warehouse. This would imply that the auto-suspend timer is affected by the number of clusters running, which is not the case. The autosuspend timer is only affected by the activity on the warehouse, regardless of the number of clusters running.
- D. Auto-suspend does apply for multi-cluster warehouses, as explained above. It is a feature that can be enabled or disabled for any warehouse, regardless of the number of clusters.

Question: 28

A company has set up a new Snowflake account. An Identity Provider (IdP) has been configured for both Single Sign-On (SSO) and SCIM provisioning.

What maintenance is required to ensure that the SCIM provisioning process continues to operate without errors?

A. The IdP service account requires a new RSA key pair to be generated every six months.

B. The Administrator must issue a POST RENEW call to the REST API at least once every six months.

C. The OAuth Bearer Tokens have a lifespan of six months and must be regenerated prior to expiration.

D. The IdP Administrator must issue a REFRESH transaction at least once every six months to synchronize all users and roles.

Answer: C

Explanation:

According to the Snowflake documentation1, the authentication process for SCIM provisioning uses an OAuth Bearer token and this token is valid for six months. Customers must keep track of their authentication token and can generate a new token on demand. If the token expires, the SCIM provisioning process will fail. Therefore, the token must be regenerated before it expires. The other options are not required for SCIM provisioning.

Question: 29

What are benefits of creating and maintaining resource monitors in Snowflake? (Select THREE).

A. The cost of running a resource monitor is only 10% of a credit, per day of operation.

B. Multiple resource monitors can be applied to a single virtual warehouse.

C. Resource monitors add no additional load to virtual warehouse compute.

D. Multiple triggers can be configured across various virtual warehouse thresholds.

E. Resource monitor governance is tightly controlled and monitors can only be created by the ACCOUNTADMIN role or users with the CREATE MONITOR privilege.

F. Resource monitors can be applied to more than one virtual warehouse.

Answer: CDF

Explanation:

According to the Snowflake documentation1, resource monitors are a feature that helps you manage and control Snowflake costs by monitoring and setting limits on your compute resources. Resource monitors do not consume any credits or add any load to the virtual warehouses they monitor1. Resource monitors can also have multiple triggers that specify different actions (such as suspending or notifying) when certain percentages of the credit quota are reached2. Resource monitors can be applied to either the entire account or a specific set of individual warehouses1. The other options are not benefits of resource monitors. The cost of running a resource monitor is negligible, not 10% of a credit3. Multiple resource monitors cannot be applied to a single virtual warehouse; only one resource monitor can be assigned to a warehouse at a time2. Resource monitor governance is not tightly controlled; account administrators can enable users with other roles to view and modify resource monitors using SQL2.

Question: 30

DatabaseA has a single schema called Schema1. This schema contains many tables and views. The ANALYST role has privileges to select from all objects in DatabaseA. Schema1. The SYSADMIN role clones DatabaseA to DatabaseA_clone.

What privileges does the ANALYST role have on tables and views in DatabaseA_clone? (Select TWO).

A. USAGE on the schema DatabaseA clone

B. USAGE on the database DatabaseA_clone. Schemal

C. SELECT on all tables, and only non-secure views in DatabaseA_clone. Schemal

D. SELECT on all tables, and only secure views in DatabaseA_clone. Schemal

E. SELECT on all tables and views in DatabaseA_clone. Schema1

Answer: CE

Explanation:

According to the Snowflake documentation, when a database or schema is cloned, the clone inherits all granted privileges on the clones of all child objects contained in the source object, such as tables and views. However, the clone of the container itself does not inherit the privileges granted on the source container. Therefore, the ANALYST role will have SELECT privilege on all tables and views in DatabaseA_clone.Schema1, but not USAGE privilege on the database or schema. The type of view (secure or non-secure) does not affect the cloning of privileges.

Question: 31

An Administrator has a warehouse which is intended to have a credit quota set for 3000 for each calendar year. The Administrator needs to create a resource monitor that will perform the following tasks:

1. At 80% usage notify the account Administrators.
2. At 100% usage suspend the warehouse and notify the account Administrators.
3. At 120% stop all running executions, suspend the warehouse, and notify the account Administrators.

Which SQL command will meet these requirements?

A. create or replace resource monitor RM1 with credit_quota=3000
start_timestamp = '2022-01-01 00:00 CET'
triggers on 80 percent do notify
on 100 percent do suspend
on 120 percent do suspend_immediate;

alter warehouse WH1 set resource_monitor = RM1;

B. create or replace resource monitor RM1 with credit_quota=3000
frequency = yearly
start_timestamp = '2022-01-01 00:00 CET'
triggers on 80 percent do notify
on 100 percent do suspend
on 120 percent do suspend_immediate;

alter warehouse WH1 set resource monitor = RM1;

C. create or replace resource monitor RM1 with credit_quota=3000 start_timestamp = '2022-01-01 00:00 CET' triggers on 80 percent do notify on 100 percent do notify and suspend
on 120 percent do notify and suspend_immediate;

alter warehouse WH1 set resource monitor = RM1;

D. create or replace resource monitor RM1 with credit_quota=3000 frequency = yearly
triggers on 80 percent do notify
on 100 percent do suspend
on 120 percent do suspend_immediate;

alter warehouse WH1 set resource_monitor = RM1;

Answer: B

Explanation:

Option B is the correct SQL command to create a resource monitor that meets the requirements. It sets the credit quota to 3000, the frequency to yearly, the start timestamp to January 1, 2022, and the triggers to notify and suspend the warehouse at the specified thresholds. Option A is incorrect because it does not specify the frequency. Option C is incorrect because it does not specify the frequency and it uses notify and suspend instead of suspend and suspend_immediate. Option D is incorrect because it does not specify the start timestamp. For more information about resource monitors, see Working with Resource Monitors and CREATE RESOURCE MONITOR.

Question: 32

A data provider wants to share data from multiple databases with a data consumer account.

How can this be accomplished?

A. The data provider needs to create a secure view and grant the USAGE privilege on each database referenced by the secure view.

B. The data provider needs to create a secure view and grant the REFERENCE_USAGE privilege on each database referenced by the secure view.

C. The data provider needs to create a secure view and grant the REFERENCE_USAGE privilege to a database role to include objects from multiple databases in a share

D. The data provider needs to create a secure view and must grant the REFERENCE_USAGE privilege on the database where the secure view is created.

Answer: B

Explanation:

Option B is the correct answer because it follows the steps described in the Snowflake documentation for sharing data from multiple databases using secure views. The data provider needs to grant the REFERENCE_USAGE privilege on each database that contains objects referenced by the secure view, and the USAGE privilege only on the database where the secure view is created. Option A is incorrect because it grants the USAGE privilege instead of the REFERENCE_USAGE privilege. Option C is incorrect because it grants the REFERENCE_USAGE privilege to a database role, which is not supported. Option D is incorrect because it grants the REFERENCE_USAGE privilege on the wrong database.

Question: 33

A requirement has been identified to allow members of a corporate Data Product team to bring in data sets from the Snowflake Marketplace. The members of this team use the role DP_TEAM.

What grant statements must the ACCOUNTADMIN execute in order for the DP TEAM role to import and work with data from the Marketplace?

A. grant marketplace in account to role dp_team;

grant create database from share to role dp_team;

B. grant usage on snowflake_marketplace to role dp_team;

grant create database on account to role dp_team;

C. grant imported privileges on account to role dp_team;

grant create database on account to role dp_team;

D. grant import share on account to role dp_team;

grant create database on account to role dp_team;

Answer: D

Explanation:

Option D is the correct answer because it follows the steps described in the Snowflake documentation for importing data from the Snowflake Marketplace. The ACCOUNTADMIN role needs to grant the IMPORT SHARE privilege on the account to the DP_TEAM role, which allows the role to

import data from any provider in the marketplace. The ACCOUNTADMIN role also needs to grant the CREATE DATABASE privilege on the account to the DP_TEAM role, which allows the role to create a database from a share. Option A is incorrect because there is no MARKETPLACE privilege in Snowflake. Option B is incorrect because the USAGE privilege on SNOWFLAKE_MARKETPLACE is not sufficient to import data from the marketplace. Option C is incorrect because there is no IMPORTED PRIVILEGES privilege in Snowflake.

Question: 34

The following commands were executed:

Grant usage on database PROD to role PROD_ANALYST;
Grant usage on database PROD to role PROD_SUPERVISOR;

Grant ALL PRIVILEGES on schema PROD. WORKING to role PROD_ANALYST;
Grant ALL PRIVILEGES on schema PROD. WORKING to role PROD_SUPERVISOR;

Grant role PROD ANALYST to user A;
Grant role PROD SUPERVISOR to user B;

What authority does each user have on the WORKING schema?

A. Only user B can create tables, because all privileges were transferred to PROD_SUPERVISOR.

B. Tables created by either user A or user B will be visible to both users.

C. All existing tables in schema PROD. WORKING will be visible to both users.

D. Both user A and user B can create tables in the PROD. WORKING schema.

Answer: D

Explanation:

Question: 35

An Administrator receives data from a Snowflake partner. The partner is sharing a dataset that contains multiple secure views. The Administrator would like to configure the data so that only certain roles can see certain secure views.

How can this be accomplished?

A. Apply RBAC directly onto the partner's shared secure views.

B. Individually grant imported privileges onto the schema in the share.

C. Clone the data and insert it into a company-owned share and apply the desired RBAC on the new tables.

D. Create views over the incoming shared database and apply the desired RBAC onto these views.

Answer: D

Explanation:

According to the Snowflake documentation1, secure views are only exposed to authorized users who have been granted the role that owns the view. Therefore, applying RBAC directly onto the partner's shared secure views (option A) is not possible, as the administrator does not own those views. Individually granting imported privileges onto the schema in the share (option B) is also not feasible, as the privileges granted on the schema do not apply to existing secure views, only to future ones2. Cloning the data and inserting it into a company-owned share (option C) is not recommended, as it would create unnecessary duplication of data and increase storage costs. The best option is to create views over the incoming shared database and apply the desired RBAC onto these views (option D). This way, the administrator can control the access to the data based on the roles in their account, without modifying the original data or views from the partner.

Question: 36

What are characteristics of data replication in Snowflake? (Select THREE).

A. The ALTER DATABASE ... ENABLE REPLICATION TO ACCOUNTS command must be issued from the primary account.

B. Users must be granted REPLICATIONADMIN privileges in order to enable replication.

C. To start replication run the ALTER DATABASE ... REFRESH command on the account where the secondary database resides.

D. Replication can only occur within the same cloud provider.

E. Databases created from shares can be replicated.

F. Users can have unlimited primary databases and they can be replicated to an unlimited number of accounts if all accounts are within the same organization.

Answer: AEF

Explanation:

- Option A is correct because the ALTER DATABASE … ENABLE REPLICATION TO ACCOUNTS command must be issued from the primary account that owns the database to be replicated1.
- Option B is incorrect because users must be granted REPLICATIONGRANTER privileges in order to enable replication1.
- Option C is incorrect because to start replication, the ALTER DATABASE … REFRESH command must be run on the primary database, not the secondary database1.
- Option D is incorrect because replication can occur across different cloud providers, as well as across regions2.
- Option E is correct because databases created from shares can be replicated, as long as the share is active and the database is not dropped or altered1.
- Option F is correct because users can have unlimited primary databases and they can be

replicated to an unlimited number of accounts if all accounts are within the same organization1.

Question: 37

An Administrator has a user who needs to be able to suspend and resume a task based on the current virtual warehouse load, but this user should not be able to modify the task or start a new run.

What privileges should be granted to the user to meet these requirements? (Select TWO).

A. EXECUTE TASK on the task

B. OWNERSHIP on the task

C. OPERATE on the task

D. USAGE on the database and schema containing the task

E. OWNERSHIP on the database and schema containing the task

Answer: CD

Explanation:

The user needs the OPERATE privilege on the task to suspend and resume it, and the USAGE privilege on the database and schema containing the task to access it1. The EXECUTE TASK privilege is not required for suspending and resuming a task, only for triggering a new run1. The OWNERSHIP privilege on the task or the database and schema would allow the user to modify or drop the task, which is not desired.

Question: 38

A Snowflake organization MYORG consists of two Snowflake accounts:

Account Name	Snowflake Region	Snowflake Edition
ACCOUNT1	AWS_EU_WEST_2	ENTERPRISE
ACCOUNT2	AZURE_WESTEUROPE	STANDARD

The ACCOUNT1 has a database PROD_DB and the ORGADMIN role enabled.

Management wants to have the PROD_DB database replicated to ACCOUNT2.

Are there any necessary configuration steps in ACCOUNT1 before the database replication can be configured and initiated in ACCOUNT2?

A. USE ROLE ORGADMIN;
SELECT SYSTEMSGLOBAL_ACCOUNT_SET_PARAMETER ('MYORG. ACCOUNT1',
'ENABLE_ACCOUNT_DATABASE_REPLICATION', 'TRUE');

SELECT SYSTEMSGLOBAL_ACCOUNT_SET_PARAMETER ('MYORG. ACCOUNT2',
'ENABLE_ACCOUNT_DATABASE_REPLICATION', 'TRUE');

USE ROLE ACCOUNTADMIN;
ALTER DATABASE PROD DB ENABLE REPLICATION TO ACCOUNTS MYORG. ACCOUNT2;

B. USE ROLE ORGADMIN;
SELECT SYSTEMSGLOBAL ACCOUNT SET_PARAMETER ('MYORG. ACCOUNT1',
'ENABLE_ACCOUNT_DATABASE_REPLICATION', 'TRUE');

USE ROLE ACCOUNTADMIN;
ALTER DATABASE PROD_DB ENABLE REPLICATION TO ACCOUNTS MYORG. ACCOUNT2 IGNORE EDITION CHECK;

C. No configuration steps are necessary in ACCOUNT1. Replicating databases across accounts within the same Snowflake organization is enabled by default.

D. It is not possible to replicate a database from an Enterprise edition Snowflake account to a Standard edition Snowflake account.

Answer: B

Explanation:

According to the Snowflake documentation1, database replication across accounts within the same organization requires the following steps:

- Link the accounts in the organization using the ORGADMIN role.
- Enable account database replication for both the source and target accounts using the SYSTEM$GLOBAL_ACCOUNT_SET_PARAMETER function.
- Promote a local database to serve as the primary database and enable replication to the target accounts using the ALTER DATABASE … ENABLE REPLICATION TO ACCOUNTS command.
- Create a secondary database in the target account using the CREATE DATABASE … FROM SHARE command.
- Refresh the secondary database periodically using the ALTER DATABASE … REFRESH command.

Option A is incorrect because it does not include the step of creating a secondary database in the target account. Option C is incorrect because replicating databases across accounts within the same organization is not enabled by default, but requires enabling account database replication for both the source and target accounts. Option D is incorrect because it is possible to replicate a database from an Enterprise edition Snowflake account to a Standard edition Snowflake account, as long as the IGNORE EDITION CHECK option is used in the ALTER DATABASE … ENABLE REPLICATION TO ACCOUNTS command2. Option B is correct because it includes all the necessary configuration steps in ACCOUNT1, except for creating a secondary database in ACCOUNT2, which can be done after the replication is enabled.

Question: 39

What is a characteristic of Snowflake's transaction locking and concurrency modeling?

A. A deadlock cannot occur in Snowflake, since concurrently executed queries and DML statements do not block one another.

B. If two queries are concurrently executed against the same table, one of the two queries will be blocked until the other query completes.

C. Transaction locking in Snowflake is enforced exclusively at the row and table levels.

D. Queries executed within a given transaction see that transaction's uncommitted changes.

Answer: A

Explanation:

According to the Snowflake documentation1, Snowflake uses a multi-version concurrency control (MVCC) model, which means that each transaction operates on a consistent snapshot of the database at a point in time. This allows queries and DML statements to run concurrently without blocking each other, as they do not modify the same data. Therefore, a deadlock, which occurs when concurrent transactions are waiting on resources that are locked by each other, cannot happen in Snowflake. Option B is incorrect because queries and DML statements do not block each other in Snowflake, unless they are explicitly started transactions and multiple statements in each transaction2. Option C is incorrect because transaction locking in Snowflake is enforced at the partition level, not the row or table level3. Option D is incorrect because queries executed within a given transaction do not see that transaction's uncommitted changes, but only the committed changes that occurred before the transaction started1.

Question: 40

A Snowflake Administrator created a role ROLE_MANAGED_ACCESS and a schema SCHEMA_MANAGED_ACCESS as follows:

```
USE ROLE SECURITYADMIN;
CREATE ROLE ROLE_MANAGED_ACCESS;
GRANT ROLE ROLE_MANAGED_ACCESS TO ROLE SYSADMIN;

GRANT USAGE ON WAREHOUSE COMPUTE_WH TO ROLE ROLE_MANAGED_ACCESS;

GRANT ALL privileges ON DATABASE WORK TO ROLE ROLE_MANAGED_ACCESS;

USE ROLE ROLE_MANAGED_ACCESS;
CREATE SCHEMA SCHEMA_MANAGED_ACCESS WITH MANAGED ACCESS;

USE ROLE SECURITYADMIN;
GRANT SELECT, INSERT ON FUTURE TABLES IN SCHEMA SCHEMA MANAGED ACCESS to
ROLE_MANAGED_ACCESS;
```

The Administrator now wants to disable the managed access on the schema.

How can this be accomplished?

A. ALTER SCHEMA SCHEMA MANAGED ACCESS DISABLE MANAGED ACCESS;

B. USE ROLE ROLE MANAGED_ACCESS;
DROP SCHEMA WORK. SCHEMA_MANAGED_ACCESS;
CREATE SCHEMA SCHEMA_MANAGED_ACCESS;
Then recreate all needed objects.

C. REVOKE SELECT, INSERT ON FUTURE TABLES IN SCHEMA SCHEMA_MANAGED_ACCESS FROM ROLE_MANAGED_ACCESS;
ALTER SCHEMA SCHEMA MANAGED ACCESS DISABLE MANAGED ACCESS;

D. USE ROLE ROLE_MANAGED_ACCESS;
DROP SCHEMA WORK. SCHEMA MANAGED_ACCESS;
CREATE SCHEMA SCHEMA_MANAGED_ACCESS WITHOUT MANAGED ACCESS;
Then recreate all needed objects.

Answer: A

Explanation:

According to the Snowflake documentation1, you can change a managed access schema to a regular schema using the ALTER SCHEMA statement with the DISABLE MANAGED ACCESS keywords. This will disable the managed access feature on the schema and revert the access control to the default behavior. Option B is incorrect because dropping and recreating the schema will also delete all the objects and metadata in the schema, which is not necessary to disable the managed access. Option C is incorrect because revoking the privileges on the future tables from the role is not required to disable the managed access. Option D is incorrect because there is no WITHOUT MANAGED ACCESS option in the CREATE SCHEMA statement.

Question: 41

Which commands can be performed by a user with the ORGADMIN role but not the ACCOUNTADMIN role? (Select TWO).

A. SHOW REGIONS;

B. SHOW USERS;

C. SHOW ORGANIZATIONACCOUNTS;

D. GRANT ROLE ORGADMIN TO USER <username>;

E. SELECT SYSTEM$GLOBAL_ACCOUNT_SET_PARAMETER ('ACCOUNT LOCATOR', 'ENABLE ACCOUNT DATABASE_REPLICATION', 'true'

);

Answer: CE

Explanation:

According to the Snowflake documentation1, the ORGADMIN role is a special system role that is responsible for managing operations at the organization level, such as creating and viewing accounts, enabling database replication, and setting global account parameters. The ACCOUNTADMIN role is a system role that is responsible for managing operations at the account level, such as creating and managing users, roles, warehouses, databases, and shares. Therefore, the commands that can be performed by the ORGADMIN role but not the ACCOUNTADMIN role are:

- SHOW ORGANIZATION ACCOUNTS: This command lists all the accounts in the organization and their properties, such as region, edition, and status2. The ACCOUNTADMIN role can only show the current account and its properties using the SHOW ACCOUNTS command3.
- SELECT SYSTEM$GLOBAL_ACCOUNT_SET_PARAMETER: This function sets a global account parameter for an account in the organization, such as enabling account database replication4. The ACCOUNTADMIN role can only set local account parameters using the ALTER ACCOUNT command. Option A is incorrect because the SHOW REGIONS command can be executed by any role, not just the ORGADMIN role. Option B is incorrect because the SHOW USERS command can be executed by the ACCOUNTADMIN role, as well as any role that has been granted the MONITOR privilege on the account. Option D is incorrect because the GRANT ROLE ORGADMIN TO USER <username> command can be executed by the ACCOUNTADMIN role, as well as any role that has been granted the ORGADMIN role1.

Question: 42

Which command can temporarily disable Multi-factor Authentication (MFA) for the Snowflake username user1 for 24 hours?

A. alter user userl set MINS_TO_BYPASS_MFA=1440;

B. alter user userl set DISABLE_MFA=1440;

C. alter user userl set TEMPORARY_MFA_BYPASS=1440;

D. alter user userl set HOURS_TO_BYPASS_MFA=24;

Answer: A

Explanation:

According to the Snowflake documentation1, the MINS_TO_BYPASS_MFA property specifies the number of minutes to temporarily disable MFA for a user so that they can log in without the temporary token generated by the Duo Mobile application. After the time passes, MFA is enforced and the user cannot log in without the token. Therefore, to disable MFA for 24 hours, the value of this property should be set to 1440 minutes (24 x 60). Option B is incorrect because the DISABLE_MFA property is a boolean value that permanently disables MFA for a user, not a numeric value that specifies the duration. Option C is incorrect

because there is no such property as TEMPORARY_MFA_BYPASS in Snowflake. Option D is incorrect because there is no such property as HOURS_TO_BYPASS_MFA in Snowflake.

Question: 43

A Snowflake Administrator wants to create a virtual warehouse that supports several dashboards, issuing various queries on the same database.

For this warehouse, why should the Administrator consider setting AUTO_SUSPEND to 0 or NULL?

A. To save costs on warehouse shutdowns and startups for different queries

B. To save costs by running the warehouse as little as possible

C. To keep the data cache warm to support good performance of similar queries

D. To keep the query result cache warm for good performance on repeated queries

Answer: C

Explanation:

According to the Snowflake documentation1, the AUTO_SUSPEND property specifies the number of seconds of inactivity after which a warehouse is automatically suspended. If the property is set to 0 or NULL, the warehouse never suspends automatically. For a warehouse that supports several dashboards, issuing various queries on the same database, setting AUTO_SUSPEND to 0 or NULL can help to keep the data cache warm, which means that the data used by the queries is already loaded into the warehouse memory and does not need to be fetched from the storage layer. This can

improve the performance of similar queries that access the same data. Option A is incorrect because setting AUTO_SUSPEND to 0 or NULL does not save costs on warehouse shutdowns and startups, but rather increases the costs by keeping the warehouse running continuously. Option B is incorrect because setting AUTO_SUSPEND to 0 or NULL does not run the warehouse as little as possible, but rather runs the warehouse as much as possible. Option D is incorrect because setting AUTO_SUSPEND to 0 or NULL does not affect the query result cache, which is a separate cache that stores the results of previous queries for a period of time. The query result cache is not dependent on the warehouse state, but on the query criteria2.

Question: 44

What SCIM integration types are supported in Snowflake? (Select THREE).

A. Amazon Web Services (AWS)

B. Google Cloud Platform (GCP)

C. Okta

D. Custom

E. Azure Active Directory (Azure AD)

F. Duo Security Provisioning Connector

Answer: CDE

Explanation:

According to the Snowflake documentation1, Snowflake supports SCIM 2.0 to integrate Snowflake with Okta and Microsoft Azure AD, which both function as identity providers. Snowflake also supports identity providers that are neither Okta nor Microsoft Azure (i.e. Custom). Therefore, the SCIM integration types that are supported in Snowflake are Okta, Custom, and Azure AD. Option A is incorrect because Amazon Web Services (AWS) is not a SCIM identity provider. Option B is incorrect because Google Cloud Platform (GCP) is not a SCIM identity provider. Option F is incorrect because Duo Security Provisioning Connector is not a SCIM identity provider.

Question: 45

A team of developers created a new schema for a new project. The developers are assigned the role DEV_TEAM which was set up using the following statements:

```
USE ROLE SECURITYADMIN;
CREATE ROLE DEV TEAM;
GRANT USAGE, CREATE SCHEMA ON DATABASE DEV_DB01 TO ROLE DEV_TEAM;
GRANT USAGE ON WAREHOUSE DEV_WH TO ROLE DEV_TEAM;
```

Each team member's access is set up using the following statements:

```
USE ROLE SECURITYADMIN;
CREATE ROLE JDOE_PROFILE;
CREATE USER JDOE LOGIN NAME = 'JDOE' DEFAULT_ROLE='JDOE_PROFILE';
GRANT ROLE JDOE_PROFILE TO USER JDOE;
GRANT ROLE DEV_TEAM TO ROLE JDOE_PROFILE;
```

New tables created by any of the developers are not accessible by the team as a whole.

How can an Administrator address this problem?

A. Assign ownership privilege to DEV_TEAM on the newly-created schema.

B. Assign usage privilege on the virtual warehouse DEV_WH to the role JDOE_PROFILE.

C. Set up future grants on the newly-created schemas.

D. Set up the new schema as a managed-access schema.

Answer: C

Explanation:

According to the Snowflake documentation1, future grants are a way to automatically grant privileges on future objects of a specific type that are created in a database or schema. By setting up future grants on the newly-created schemas, the administrator can ensure that any tables created by the developers in those schemas will be accessible by the DEV_TEAM role, without having to grant privileges on each table individually. Option A is incorrect because assigning ownership privilege to DEV_TEAM on the newly-created schema does not grant privileges on the tables in the schema, only on the schema itself. Option B is incorrect because assigning usage privilege on the virtual warehouse DEV_WH to the role JDOE_PROFILE does not affect the access to the tables in the schemas, only the ability to use the warehouse. Option D is incorrect because setting up the new schema as a managed-access schema does not grant privileges on the tables in the schema, but rather requires explicit grants for each table.

Question: 46

A large international company with many operating regions requires data to be shared bidirectionally among all offices (head office to regional offices and regional offices among themselves). This company is a Snowflake account holder with European operations deployed in Microsoft Azure (single region) while North American regional offices are using AWS (single region) as their deployment cloud. This setup is required to comply with Personal Identifiable Information (PII) regulations in some of the European countries. The corporate head office is in Europe.

How can this data be shared bi-directionally, while MINIMIZING costs?

A. Use data replication everywhere to reduce costs associated with same-region sharing.

B. Use the PUT command to move files to an Amazon S3 bucket and Azure Blobs, and use an external file management application to move files within the corporate VPC.

C. Move all the Snowflake accounts to a single region, and implement data sharing.

D. Use bi-directional data sharing among offices in the same region and replication among offices across the continents.

Answer: D

Explanation:

According to the Snowflake documentation1, data sharing is a feature that allows sharing selected objects in a database in one account with other accounts in the same organization, without copying or transferring any data. Data sharing is supported across regions and across cloud platforms, but it requires enabling account

database replication for both the source and target accounts2. Data replication is a feature that allows replicating objects from a source account to one or more target accounts in the same organization, providing read-only access for the replicated objects. Data replication is also supported across regions and across cloud platforms, but it incurs additional storage costs for the replicated data2. Therefore, the best way to share data bi-directionally among all offices, while minimizing costs, is to use data sharing among offices in the same region, which does not require replication or additional storage, and use replication among offices across the continents, which provides near real-time access to the shared data. Option A is incorrect because using data replication everywhere would increase the costs associated with additional storage and compute resources for the replicated data. Option B is incorrect because using the PUT command to move files to an Amazon S3 bucket and Azure Blobs, and using an external file management application to move files within the corporate VPC, would not leverage the benefits of Snowflake's data sharing and replication features, and would also incur additional costs and complexity for data transfer and synchronization. Option C is incorrect because moving all the Snowflake accounts to a single region would violate the PII regulations in some of the European countries, and would also incur additional costs and complexity for data migration and consolidation.

Question: 47

What are benefits of using Snowflake organizations? (Select TWO).

A. Administrators can change Snowflake account editions on-demand based on need.

B. Administrators can monitor and understand usage across all accounts in the organization.

C. Administrators can simplify data movement across all accounts within the organization.

D. User administration is simplified across all accounts within the organization.

E. Administrators have the ability to create accounts in any available cloud provider or region.

Answer: BE

Explanation:

According to the Snowflake documentation1, organizations are a feature that allows linking the accounts owned by a business entity, simplifying account management and billing, replication and failover, data sharing, and other account administration tasks. Some of the benefits of using organizations are:

- Administrators can monitor and understand usage across all accounts in the organization using the ORGANIZATION_USAGE schema, which provides historical usage data for all accounts in the organization via views in a shared database named SNOWFLAKE2. This can help to optimize costs and performance across the organization.
- Administrators have the ability to create accounts in any available cloud provider or region using the CREATE ACCOUNT command, which allows specifying the cloud platform and region for the new account3. This can help to meet the business needs and compliance requirements of the organization.

Option A is incorrect because administrators cannot change Snowflake account editions on-demand based on need, but rather have to contact Snowflake Support to request an edition change4. Option C is incorrect because administrators cannot simplify data movement across all accounts within the organization, but rather

have to enable account database replication for both the source and target accounts, and use the ALTER DATABASE … ENABLE REPLICATION TO ACCOUNTS command to promote a local database to serve as the primary database and enable replication to the target accounts5. Option D is incorrect because user administration is not simplified across all accounts within the organization, but rather requires creating and managing users, roles, and privileges for each account separately, unless using a federated authentication method such as SSO or SCIM.

Question: 48

A Snowflake account is configured with SCIM provisioning for user accounts and has bi-directional synchronization for user identities. An Administrator with access to SECURITYADMIN uses the Snowflake UI to create a user by issuing the following commands:

```
use role USERADMIN;
create or replace role DEVELOPER_ROLE;

create user PTORRES PASSWORD = 'hello world!' MUST_CHANGE_PASSWORD = FALSE default_role = DEVELOPER_ROLE;
```

The new user named PTORRES successfully logs in, but sees a default role of PUBLIC in the web UI. When attempted, the following command fails:

```
use DEVELOPER_ROLE;
```

Why does this command fail?

A. The DEVELOPER_ROLE needs to be granted to SYSADMIN before user PTORRES will be able to use the role.

B. The new role can only take effect after USERADMIN has logged out.

C. USERADMIN needs to explicitly grant the DEVELOPER_ROLE to the new USER.

D. The new role will only take effect once the identity provider has synchronized by way of SCIM with the Snowflake account.

Answer: C

Explanation:

According to the Snowflake documentation1, creating a user with a default role does not automatically grant that role to the user. The user must be explicitly granted the role by the role owner or a higher-level role. Therefore, the USERADMIN role, which created the DEVELOPER_ROLE, needs to explicitly grant the DEVELOPER_ROLE to the new user PTORRES using the GRANT ROLE command. Otherwise, the user PTORRES will not be able to use the DEVELOPER_ROLE and will see the default role of PUBLIC in the web UI. Option A is

incorrect because the DEVELOPER_ROLE does not need to be granted to SYSADMIN before user PTORRES can use the role. Option B is incorrect because the new role can take effect immediately after it is created and granted to the user, and does not depend on the USERADMIN role logging out. Option D is incorrect because the new role will not be affected by the identity provider synchronization, as it is created and managed in Snowflake.

Question: 49

Which type of listing in the Snowflake Marketplace can be added and queried immediately?

A. Monetized listing
B. Standard listing
C. Regional listing
D. Personalized listing

Answer: B

Explanation:

According to the Snowflake documentation1, a standard listing is a type of listing that provides free access to the full data product, with no payment required. A standard listing can be added and queried immediately by the consumer, as long as they accept the terms and conditions of the listing.

A monetized listing is a type of listing that charges for access to the data product, using the pricing models offered by Snowflake. A monetized listing requires the consumer to provide payment information and agree to the billing terms before accessing the data product. A regional listing is not a type of listing, but a way to specify the regions where the listing is available. A personalized listing is a type of listing that provides limited trial access to the data product, with unlimited access to the full data product available upon request. A personalized listing requires the consumer to request access from the provider and wait for the provider to grant access before accessing the data product. Therefore, the only type of listing that can be added and queried immediately is the standard listing.

Question: 50

A virtual warehouse report_wh is configured with AUTO_RESUME=TRUE and AUTO_SUSPEND=300. A user has been granted the role accountant.

An application with the accountant role should use this warehouse to run financial reports, and should keep track of compute credits used by the warehouse.

What minimal privileges on the warehouse should be granted to the role to meet the requirements for the application? (Select TWO).

A. OPERATE

B. MODIFY

C. MONITOR

D. USAGE

E. OWNERSHIP

Answer: CD

Explanation:

According to the Snowflake documentation1, the MONITOR privilege on a warehouse grants the ability to view the warehouse usage and performance metrics, such as the number of credits consumed, the average and maximum run time, and the number of queries executed. The USAGE privilege on a warehouse grants the ability to use the warehouse to execute queries and load data. Therefore, the minimal privileges on the warehouse that should be granted to the role to meet the requirements for the application are MONITOR and USAGE. Option A is incorrect because the OPERATE privilege on a warehouse grants the ability to start, stop, resume, and suspend the warehouse, which is not required for the application. Option B is incorrect because the MODIFY privilege on a warehouse grants the ability to alter the warehouse properties, such as the size, autosuspend, and auto-resume settings, which is not required for the application. Option E is incorrect because the OWNERSHIP privilege on a warehouse grants the ability to drop the warehouse, grant or revoke privileges on the warehouse, and transfer the ownership to another role, which is not required for the application.

Question: 51

What is required for stages, without credentials, to limit data exfiltration after a storage integration and associated stages are created?

A. ALTER ACCOUNT my_account SET
REQUIRE_STORAGE_INTEGRATION_FOR_STAGE_CREATION = true;

ALTER ACCOUNT my_account SET
REQUIRE_STORAGE_INTEGRATION_FOR_STAGE_OPERATION = true;

ALTER ACCOUNT my_account SET
PREVENT_UNLOAD_TO_INLINE_URL = false;

B. ALTER ACCOUNT my_account SET
REQUIRE_STORAGE_INTEGRATION_FOR_STAGE_CREATION = false;

ALTER ACCOUNT my_account SET
REQUIRE_STORAGE_INTEGRATION_FOR_STAGE_OPERATION = false;

ALTER ACCOUNT my_account SET
PREVENT_UNLOAD_TO_INLINE_URL = true;

C. ALTER ACCOUNT my_account SET
REQUIRE_STORAGE_INTEGRATION_FOR_STAGE_CREATION = false;

ALTER ACCOUNT my_account SET
REQUIRE_STORAGE_INTEGRATION_FOR_STAGE_OPERATION = false;

ALTER ACCOUNT my_account SET
PREVENT_UNLOAD_TO_INLINE_URL = false;

D. ALTER ACCOUNT my_account SET
REQUIRE_STORAGE_INTEGRATION_FOR_STAGE_CREATION = true;

ALTER ACCOUNT my_account SET
REQUIRE_STORAGE_INTEGRATION FOR STAGE_OPERATION = true;

ALTER ACCOUNT my_account SET
PREVENT_UNLOAD_TO_INLINE_URL = true;

Answer: D

Explanation:

According to the Snowflake documentation1, stages without credentials are a way to create external stages that use storage integrations to access data files in cloud storage without providing any credentials to Snowflake. Storage integrations are objects that define a trust relationship between Snowflake and a cloud provider, allowing Snowflake to authenticate and authorize access to the cloud storage. To limit data exfiltration after a storage integration and associated stages are created, the following account-level parameters can be set:

- REQUIRE_STORAGE_INTEGRATION_FOR_STAGE_CREATION: This parameter enforces that all external stages must be created using a storage integration. This prevents users from creating external stages with inline credentials or URLs that point to unauthorized locations.
- REQUIRE_STORAGE_INTEGRATION_FOR_STAGE_OPERATION: This parameter enforces that all operations on external stages, such as PUT, GET, COPY, and LIST, must use a storage integration. This prevents users from performing operations on external stages with inline credentials or URLs that point to unauthorized locations.
- PREVENT_UNLOAD_TO_INLINE_URL: This parameter prevents users from unloading data from Snowflake tables to inline URLs that do not use a storage integration. This prevents users from exporting data to unauthorized locations.

Therefore, the correct answer is option D, which sets all these parameters to true. Option A is incorrect because it sets PREVENT_UNLOAD_TO_INLINE_URL to false, which allows users to unload data to inline URLs that do not use a storage integration. Option B is incorrect because it sets both REQUIRE_STORAGE_INTEGRATION_FOR_STAGE_CREATION and REQUIRE_STORAGE_INTEGRATION_FOR_STAGE_OPERATION to false, which allows users to create and operate on external stages without using a storage integration. Option C is incorrect because it sets all the parameters to false, which does not enforce any restrictions on data exfiltration.

Question: 52

A Snowflake Administrator has a multi-cluster virtual warehouse and is using the Snowflake Business Critical edition. The minimum number of clusters is set to 2 and the maximum number of clusters is set to 10. This configuration works well for the standard workload, rarely exceeding 5 running clusters. However, once a month the Administrator notes that there are a few complex long-running queries that are causing increased queue time and the warehouse reaches its maximum limit at 10 clusters.

Which solutions will address the issues happening once a month? (Select TWO).

A. Use a task to increase the cluster size for the time period that the more complex queries are running and another task to reduce the size of the cluster once the complex queries complete.

B. Have the group running the complex monthly queries use a separate appropriately-sized warehouse to support their workload.

C. Increase the multi-cluster maximum to 20 or more clusters.

D. Examine the complex queries and determine if they can be made more efficient using clustering keys or materialized views.

E. Increase the minimum number of clusters started in the multi-cluster configuration to 5.

Answer: AB

Explanation:

According to the Snowflake documentation1, a multi-cluster warehouse is a virtual warehouse that consists of multiple clusters of compute resources that can scale up or down automatically to handle the concurrency and performance needs of the queries submitted to the warehouse. A multi-cluster warehouse has a minimum and maximum number of clusters that can be specified by the administrator. Option A is a possible solution to address the issues happening once a month, as it allows the administrator to use a task to increase the cluster size for the time period that the more complex queries are running and another task to reduce the size of the cluster once the complex queries complete. This way, the warehouse can have more resources available to handle the complex queries without reaching the maximum limit of 10 clusters, and then return to the normal cluster size to save costs. Option B is another possible solution to address the issues happening once a month, as it allows the administrator to have the group running the complex monthly queries use a separate appropriately-sized warehouse to support their workload. This way, the warehouse can isolate the complex queries from the standard workload and avoid queue time and resource contention. Option C is not a recommended solution to address the issues happening once a month, as it would increase the costs and complexity of managing the multi-cluster warehouse, and may not solve the underlying problem of inefficient queries. Option D is a good practice to improve the performance of the queries, but it is not a direct solution to address the issues happening once a month, as it requires analyzing and optimizing the complex queries using clustering keys or materialized views, which may not be feasible or effective in all cases. Option E is not a recommended solution to address the issues happening once a month, as it would increase the costs and waste resources by starting more clusters than needed for the standard workload.

Question: 53

A company has many users in the role ANALYST who routinely query Snowflake through a reporting tool. The Administrator has noticed that the ANALYST users keep two small clusters busy all of the time, and occasionally they need three or four clusters of that size.

Based on this scenario, how should the Administrator set up a virtual warehouse to MOST efficiently support this group of users?

A. Create a multi-cluster warehouse with MIN_CLUSTERS set to 1. Give MANAGE privileges to the ANALYST role so this group can start and stop the warehouse, and increase the number of clusters as needed.

B. Create a multi-cluster warehouse with MIN_CLUSTERS set to 2. Set the warehouse to auto-resume and auto-suspend, and give USAGE privileges to the ANALYST role. Allow the warehouse to autoscale.

C. Create a standard X-Large warehouse, which is equivalent to four small clusters. Set the warehouse to auto-resume and auto-suspend, and give USAGE privileges to the ANALYST role.

D. Create four virtual warehouses (sized Small through XL) and set them to auto-suspend and autoresume. Have users in the ANALYST role select the appropriate warehouse based on how many

queries are being run.

Answer: B

Explanation:

According to the Snowflake documentation1, a multi-cluster warehouse is a virtual warehouse that consists of multiple clusters of compute resources that can scale up or down automatically to handle the concurrency and performance needs of the queries submitted to the warehouse. A multi-cluster warehouse has a minimum and maximum number of clusters that can be specified by the administrator. Option B is the most efficient way to support the group of users, as it allows the administrator to create a multi-cluster warehouse with MIN_CLUSTERS set to 2, which means that the warehouse will always have two clusters running to handle the standard workload. The warehouse can also auto-scale up to the maximum number of clusters (which can be set according to the peak workload) when there is a spike in demand, and then scale down when the demand decreases. The warehouse can also auto-resume and auto-suspend, which means that the warehouse will automatically start when a query is submitted and automatically stop after a period of inactivity. The administrator can also give USAGE privileges to the ANALYST role, which means that the users can use the warehouse to execute queries and load data, but not modify or operate the warehouse. Option A is not efficient, as it requires the users to manually start and stop the warehouse, and increase the number of clusters as needed, which can be time-consuming and error- prone. Option C is not efficient, as it creates a standard X-Large warehouse, which is equivalent to four small clusters, which may be more than needed for the standard workload, and may not be enough for the peak workload. Option D is not efficient, as it creates four virtual warehouses of different sizes, which can be confusing and cumbersome for the users to select the appropriate warehouse based on how many queries are being run, and may also result in wasted resources and costs.

Question: 54

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?

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.

Question: 55

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?

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: 56

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

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: 57

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?

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: 58

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

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: 59

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

A. Stages

B. Users

C. Warehouses

D. Roles

E. Shares

Answer: BD

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: 60

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

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: 61

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

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: 62

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?

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: 63

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?

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: 64

What roles can be used to create network policies within Snowflake accounts? (Select THREE).

A. SYSADMIN

B. SECURITYADMIN

C. ACCOUNTADMIN

D. ORGADMIN

E. Any role with the global permission of CREATE NETWORK POLICY

F. Any role that owns the database where the network policy is created

Answer: BCE

Explanation:

Network policies are used to restrict access to the Snowflake service and internal stages based on user IP address1. To create network policies, a role must have the global permission of CREATE NETWORK POLICY2. By default, the system-defined roles of SECURITYADMIN and ACCOUNTADMIN have this permission3. However, any other role can be granted this permission by an administrator4. Therefore, the answer is B, C, and E. The other options are incorrect because SYSADMIN and ORGADMIN do not have the CREATE NETWORK POLICY permission by default3, and network policies are not tied to specific databases5.

Question: 65

In general, the monthly billing for database replication is proportional to which variables? (Select TWO).

A. The frequency of changes to the primary database as a result of data loading or DML operations

B. The amount of table data in the primary database that changes as a result of data loading or DML operations

C. The frequency of the secondary database refreshes from the primary database

D. The number of times data moves across regions and/or cloud service providers between the primary and secondary database accounts

E. The number and size of warehouses defined in the primary account

Answer: AB

Explanation:

Snowflake charges for database replication based on two categories: data transfer and compute resources1. Data transfer costs depend on the amount of data that is transferred from the primary database to the secondary database across regions and/or cloud service providers2. Compute resource costs depend on the use of Snowflake-provided compute resources to copy data between accounts across regions1. Both data transfer and compute resource costs are proportional to the frequency and amount of changes to the primary database as a result of data loading or DML operations3. Therefore, the answer is A and B. The other options are not directly related to the replication billing, as the frequency of secondary database refreshes does not affect the amount of data transferred or copied4, and the number and size of warehouses defined in the primary account

do not affect the replication process5.

Question: 66

Which statement allows this user to access this Snowflake account from a specific IP address (192.168.1.100) while blocking their access from anywhere else?

A. CREATE NETWORK POLICY ADMIN_POLICY
ALLOWED_IP_LIST = ('192.168.1.100');
ALTER USER ABC SET NETWORK_POLICY = 'ADMIN_POLICY';

User ABC is the only user with an ACCOUNTADMIN role.

B. CREATE NETWORK POLICY ADMIN POLICY
ALLOWED_IP_LIST = ('192.168.1.100');

ALTER ROLE ACCOUNTADMIN SET NETWORK_POLICY = 'ADMIN_POLICY';

C. CREATE NETWORK POLICY ADMIN_POLICY
ALLOWED IP LIST = ('192.168.1.100')
BLOCKED_IP_LIST = ('0.0.0.0/0');
ALTER USER ABC SET NETWORK_POLICY = 'ADMIN_POLICY';

D. CREATE OR REPLACE NETWORK POLICY ADMIN_POLICY ALLOWED_IP_LIST = ('192.168. 1. 100/0')
;
ALTER USER ABC SET NETWORK_POLICY = 'ADMIN_POLICY';

Answer: C

Explanation:
Option C creates a network policy that allows only the IP address 192.168.1.100 and blocks all other IP addresses using the CIDR notation 0.0.0.0/01. It then applies the network policy to the user ABC, who has the ACCOUNTADMIN role. This ensures that only this user can access the Snowflake account from the specified IP address, while blocking their access from anywhere else. Option A does not block any other IP addresses, option B applies the network policy to the role instead of the user, and option D uses an invalid CIDR notation.

Question: 67

Review the output of the SHOW statement below which displays the current grants on the table DB1.
S1. T1:

This statement is executed:

USE ROLE ACCOUNTADMIN;
DROP ROLE A;

What will occur?

A. The table object DB1. S1. T1 will be dropped.

B. The OWNERSHIP privilege on table DB1. S1. T1 will be transferred to the ACCOUNTADMIN role.

C. The SELECT privilege on table DB1. S1. T1 to role B will be shown as GRANTED_BY the role ACCOUNTADMIN.

D. The SELECT privileges for roles B and C will remain.

Answer: D

Explanation:

Dropping role A does not affect the SELECT privileges granted to roles B and C on the table DB1.S1.T1. According to the Snowflake documentation, dropping a role revokes all privileges granted to the role, but does not revoke any privileges granted by the role. Therefore, the OWNERSHIP privilege on the table DB1.S1.T1 will be revoked from role A, but the SELECT privileges granted by role A to role B and by role B to role C will remain. The GRANTED_BY column will still show the original grantor of the privilege, not the ACCOUNTADMIN role.

Question: 68

A company's Snowflake account has multiple roles. Each role should have access only to data that resides in the given role's specific region.

When creating a row access policy, which code snippet below will provide privileges to the role ALL_ACCESS_ROLE to see all rows regardless of region, while the other roles can only see rows for their own regions?

A. create or replace row access policy region policy as (region_value varchar) returns boolean ->

'ALL ACCESS_ROLE' = current_role ()
and exists (
select 1 from entitlement_table
where role = current_role ()
and region = region_value
)

B. create or replace row access policy region policy as (region_value varchar) returns boolean -> exists (
select 1 from entitlement_table
where role = current_role ()
and region = region_value
)

C. create or replace row access policy region policy as (region_value varchar) returns boolean ->
'ALL_ACCESS_ROLE' = current_role ()
or exists (
select 1 from entitlement_table
where role = current_role ()
and region = region_value
)

D. create or replace row access policy region policy as (region_value varchar) returns boolean -> 'ALL ACCESS ROLE' = current_role ()
)

Answer: C

Explanation:

This code snippet will create a row access policy that returns true if the current role is ALL_ACCESS_ROLE or if the current role matches the region value in the entitlement_table. This means that the ALL_ACCESS_ROLE can see all rows regardless of region, while the other roles can only see rows for their own regions. According to the Snowflake documentation, the CURRENT_ROLE context function returns the name of the current role for the session. The EXISTS function returns true if the subquery returns any rows. The OR operator returns true if either operand is true. Therefore, this code snippet satisfies the requirements of the question.

Question: 69

An Administrator needs to create a sample of the table LINEITEM. The sample should not be repeatable and the sampling function should take the data by blocks of rows.

What select command will generate a sample of 20% of the table?

A. select * from LINEITEM sample bernoulli (20);

B. select * from LINEITEM sample system (20);

C. select * from LINEITEM tablesample block (20 rows);

D. select * from LINEITEM tablesample system (20) seed (1);

Answer: B

Explanation:

This command will generate a sample of 20% of the table by using the SYSTEM (or BLOCK) sampling method, which selects each block of rows with a probability of 20/100. This method is suitable for taking data by blocks of rows, as the question requires. According to the Snowflake documentation, "SYSTEM (or BLOCK): Includes each block of rows with a probability of p/100. Similar to flipping a weighted coin for each block of rows. This method does not support fixed-size sampling." The other options are either incorrect or do not meet the requirements of the question. Option A uses the BERNOULLI (or ROW) sampling method, which selects each row with a probability of 20/100, but does not take data by blocks of rows. Option C uses the BLOCK sampling method, but specifies a fixed number of rows (20) instead of a percentage (20%). Option D uses the SYSTEM sampling method, but specifies a seed value (1), which makes the sampling repeatable, contrary to the question.

Question: 70

How should an Administrator configure a Snowflake account to use AWS PrivateLink?

A. Create CNAME records in the DNS.

B. Contact Snowflake Support.

C. Block public access to Snowflake.

D. Use SnowCD to evaluate the network connection.

Answer: A

Explanation:

To configure a Snowflake account to use AWS PrivateLink, the Administrator needs to create CNAME records in the DNS that point to the private endpoints provided by Snowflake. This allows the clients to connect to Snowflake using the same URL as before, but with private connectivity. According to the Snowflake documentation, "After you have created the VPC endpoints, Snowflake provides you with a list of private endpoints for your account. You must create CNAME records in your DNS that point to these private endpoints. The CNAME records must use the same hostnames as the original Snowflake URLs for your account." The other options are either incorrect or not sufficient to configure AWS PrivateLink. Option B is not necessary, as the Administrator can enable AWS PrivateLink using the SYSTEM$AUTHORIZE_PRIVATELINK function1. Option C is not recommended, as it may prevent some data traffic from reaching Snowflake, such as large result sets stored on AWS S32. Option D is not related to AWS PrivateLink, but to Snowflake Connectivity Diagnostic (SnowCD),

which is a tool for diagnosing network issues between clients and Snowflake3.

Question: 71

A Snowflake Administrator is investigating why a query is not re-using the persisted result cache.

The Administrator found the two relevant queries from the SNOWFLAKE. ACCOUNT_USAGE. QUERY_HISTORY view:

START.TIME	USER.NAME	ROLE.NAME	WAREHOUSE.NAME	QUERY-TEXT	EXECUTION-STA'	BY TES.SCANNED	QUERYJD
2022-11-30 01:49:09.124 -0800	USER1	A	WH.FINANCE	SELECT ■ FROM DBS1 JI WHERECREATE.DATEM CURRENT.DATEO ANDLAST.MODIFIEO 4 CURRENT.TIMESTAMPO:	SUCCESS	2,048	01a8a70d-3201-6ab7-0000-e125001
2022-11-30 01:49:19.442 -0800	USER1	8	WH.PROD	SELECT • FROM DB.S1 T1 WHERE CREATE.DATE >= CURRENT.DATEO AND LAST.MODIFIED < CURRENT.TIMESTAMPO;	SUCCESS	2,048	01a8a70d-3201-6a89-0000-e125001

Why is the second query re-scanning micro-partitions instead of using the first query's persisted result cache?

A. The second query includes a CURRENT_TIMESTAMP () function.

B. The second query includes a CURRENT_DATE () function.

C. The queries are executed with two different virtual warehouses.

D. The queries are executed with two different roles.

Answer: A

Explanation:

The inclusion of the CURRENT_TIMESTAMP() function in the second query prevents it from re-using the first query's persisted result cache because this function makes each execution unique due to the constantly changing timestamp. According to the Snowflake documentation, "The query does not include non-reusable functions, which return different results for successive runs of the same query. UUID_STRING, RANDOM, and RANDSTR are good examples of non-reusable functions." The CURRENT_TIMESTAMP() function is another example of a non-reusable function, as it returns the current date and time at the start of query execution, which varies for each run. Therefore, the second query is not identical to the first query, and the result cache is not reused. The other options are either incorrect or irrelevant to the question. Option B is incorrect, as the CURRENT_DATE() function is a reusable function, as it returns the same value for all queries executed within the same day. Option C is irrelevant, as the virtual warehouse used to execute the query does not affect the result cache reuse. Option D is also irrelevant, as the role used to execute the query does not affect the result cache reuse, as long as the role has the necessary access privileges for all the tables used in the query.

Question: 72

The ACCOUNTADMIN of Account 123 works with Snowflake Support to set up a Data Exchange. After the exchange is populated with listings from other Snowflake accounts, what roles in Account 123 are allowed to request and get data?

A. Only the ACCOUNTADMIN role, and no other roles

B. Any role with USAGE privilege on the Data Exchange

C. Any role with IMPORT SHARE and CREATE DATABASE privileges

D. Any role that the listing provider has designated as authorized

Answer: B

Explanation:

To request and get data from a Data Exchange, the role in Account 123 must have the USAGE privilege on the Data Exchange object. This privilege allows the role to view the listings and request access to the data. According to the Snowflake documentation, "To view the listings in a data exchange, a role must have the USAGE privilege on the data exchange object. To request access to a listing, a role must have the USAGE privilege on the data exchange object and the IMPORT SHARE privilege on the account." The other options are either incorrect or not sufficient to request and get data from a Data Exchange. Option A is incorrect, as the ACCOUNTADMIN role is not the only role that can request and get data, as long as other roles have the necessary privileges. Option C is incorrect, as the IMPORT SHARE and CREATE DATABASE privileges are not required to request and get data, but only to create a database from a share after the access is granted. Option D is incorrect, as the listing provider does not designate the authorized roles in Account 123, but only approves or denies the requests from Account 123.

Question: 73

What information is required from the Identity Provider (IdP) to enable federated authentication in Snowflake? (Select TWO).

A. IdP account details

B. URL endpoint for SAML requests

C. SAML response format

D. Authentication certificate

E. IdP encryption key

Answer: B, D

Explanation:

To enable federated authentication (aka SSO via SAML 2.0) in Snowflake, the integration with an Identity Provider (IdP) must be configured. This setup involves configuring external authentication via SAML, and Snowflake needs specific information from the IdP.

Required Information from IdP:

URL Endpoint for SAML Requests (B)

This is often referred to as the SSO URL or SAML 2.0 Endpoint (HTTP).

It's the URL that Snowflake redirects users to for authentication.

In Snowflake's SAML configuration, this is required as the SAML2_ISSUER or SAML2_SSO_URL.

Authentication Certificate (D)

This is the X.509 certificate issued by the IdP.

It's used by Snowflake to validate the digital signature of the SAML assertions sent by the IdP.

It ensures that the SAML response is authentic and not tampered with.

X Why Other Options Are Incorrect:

A . IdP account details

Not needed. Snowflake doesn't require credentials or internal details from the IdP. It relies on assertions sent via SAML, not stored accounts.

C . SAML response format

Snowflake adheres to SAML 2.0 standard, and expects a compliant format. There's no need to specify format explicitly — it's part of the standard protocol.

E . IdP encryption key

Not required by Snowflake. Snowflake verifies SAML assertions via signature validation, not encryption using the IdP's private key.

SnowPro Administrator Reference:

Snowflake Documentation — Federated Authentication Setup

https://docs.snowflake.com/en/user-guide/security-fed-auth-use

https://docs.snowflake.com/en/user-guide/security-fed-auth-config

Required IdP Metadata for Snowflake SAML Configuration:

SAML2_SSO_URL: SAML 2.0 POST binding endpoint

SAML2_X509_CERT: Public cert used to validate IdP signatures

Question: 74

A Snowflake Administrator needs to retrieve the list of the schemas deleted within the last two days from the DB1 database.

Which of the following will achieve this?

A. SHOW SCHEMAS IN DATABASE DB1;

B. SELECT * FROM SNOWFLAKE.ACCOUNT_USAGE.SCHEMATA;

C. SELECT * FROM DB1.INFORMATION_SCHEMA.SCHEMATA;

D. SELECT * FROM SNOWFLAKE.ACCOUNT_USAGE.DATABASES;

Answer: B

Explanation:

To retrieve a list of schemas deleted within the last 2 days from the DB1 database, you need a metadata view that includes historical data, including dropped (deleted) objects.

Let's review the options:

Q B. SNOWFLAKE.ACCOUNT_USAGE.SCHEMATA

This is the correct choice because:

It includes metadata for all schemas, even deleted ones, within the retention period.

It contains a DELETED column and a DELETED_ON timestamp column.

You can filter rows with:

sql

CopyEdit

```
SELECT *
FROM SNOWFLAKE.ACCOUNT_USAGE.SCHEMATA
WHERE DELETED IS TRUE
AND DELETED_ON >= DATEADD(DAY, -2, CURRENT_TIMESTAMP())
AND CATALOG_NAME = 'DB1';
```

X A. SHOW SCHEMAS IN DATABASE DB1;

Only shows current (active) schemas — does not include deleted schemas.

X C. DB1.INFORMATION_SCHEMA.SCHEMATA

Like option A, this view only includes active schemas in the current database.
No info on deleted schemas is retained.
X D. SNOWFLAKE.ACCOUNT_USAGE.DATABASES
This metadata view tracks databases, not individual schemas.

SnowPro Administrator Reference:
SNOWFLAKE.ACCOUNT_USAGE.SCHEMATA documentation
Metadata includes both active and deleted schemas (within retention window).

Question: 75

What Snowflake capabilities are commonly used in rollback scenarios? (Select TWO).
A. SELECT SYSTEM$CANCEL_QUERY('problematic_query_id');
B. CREATE TABLE prd_table_bkp CLONE prd_table BEFORE (STATEMENT => 'problematic_query_id');
C. CREATE TABLE prd_table_bkp AS SELECT * FROM TABLE(RESULT_SCAN('problematic_query_id'));
D. ALTER TABLE prd_table SWAP WITH prd_table_bkp;
E. Contact Snowflake Support to retrieve Fail-safe data.

Answer: B, D

Explanation:
Scenario: You want to rollback changes due to a problematic query (e.g., accidental data modification or corruption). Snowflake provides two powerful tools:

Q B. CLONE ... BEFORE (STATEMENT => 'query_id')
This uses Time Travel + Zero-Copy Cloning.
You can clone a table as it existed before a specific query.
It creates a full copy of the table's state at that moment without duplicating storage.
Example:

```
CREATE TABLE prd_table_bkp CLONE prd_table
BEFORE (STATEMENT => '01a2b3c4-0000-0000-0000-123456789abc');
```

Q D. ALTER TABLE ... SWAP WITH ...
Once you've cloned the backup, you can swap it with the live table.
This is a fast, atomic operation — ideal for rollback.
Example:

```
ALTER TABLE prd_table SWAP WITH prd_table_bkp;
```

X Why the Other Options Are Incorrect:
A . SELECT SYSTEM$CANCEL_QUERY(...)
Cancels a currently running query — doesn't help if the query already executed and caused damage.
C . CREATE TABLE ... AS SELECT * FROM RESULT_SCAN(...)
Reconstructs results, not the original table.
Only captures output rows, not full table state.
Not ideal for rollback.
E . Contact Snowflake Support to retrieve Fail-safe data
Fail-safe is for disaster recovery only, and only accessible by Snowflake support.
It's not intended for routine rollback or recovery and has a 7-day fixed retention (non-configurable).

SnowPro Administrator Reference:
Zero-Copy Cloning with Time Travel

ALTER TABLE SWAP
System Functions – SYSTEM$CANCEL_QUERY
Fail-safe Overview

Question: 76

A user accidentally truncated the data from a frequently-modified table. The Administrator has reviewed the query history and found the truncate statement which was run on 2021-12-12 15:00 with query ID 8e5d0ca9-005e-44e6-b858-a8f5b37c5726. Which of the following statements would allow the Administrator to create a copy of the table as it was exactly before the truncated statement was executed, so it can be checked for integrity before being inserted into the main table?

A. CREATE TABLE RESTORE_TABLE CLONE CURRENT_TABLE BEFORE (timestamp => '2021-12-12 00:00');
B. SELECT * FROM CURRENT_TABLE before (statement => '8e5d0ca9-005e-44e6-b858- a8f5b37c5726');
C. INSERT INTO CURRENT_TABLE SELECT * FROM CURRENT_TABLE before (statement => '8e5d0ca9- 005e-44e6-b858-a8f5b37c5726');
D. CREATE TABLE RESTORE_TABLE CLONE CURRENT_TABLE before (statement => '8e5d0ca9-005e- 44e6-b858-a8f5b37c5726');

Answer: D

Explanation:

I Scenario:
A TRUNCATE command was accidentally run on a frequently modified table.
Query ID and timestamp are known.
Goal: restore a copy of the table as it existed right before the problematic statement, without affecting the current table.
Q Why Option D is Correct:
sql
CopyEdit
CREATE TABLE RESTORE_TABLE CLONE CURRENT_TABLE
BEFORE (STATEMENT => '8e5d0ca9-005e-44e6-b858-a8f5b37c5726');
This uses Zero-Copy Cloning + Time Travel.
The BEFORE (STATEMENT => ...) clause restores the exact state of the table before the TRUNCATE ran.
Creating a clone ensures the original table remains untouched for integrity checks before merging data back.
X Why Others Are Incorrect:
A . BEFORE (timestamp => '2021-12-12 00:00')
Wrong timestamp: that's 15 hours before the truncate happened. Too early; may lose needed updates.
B . SELECT * FROM CURRENT_TABLE before (statement => …)
Syntax is invalid: SELECT can't use BEFORE (STATEMENT => ...) directly like this.
C . INSERT INTO CURRENT_TABLE SELECT * FROM CURRENT_TABLE before (statement => …)
Same syntax issue. Also risky — directly inserting into the original table without validating the data first.

SnowPro Administrator Reference:
Cloning with Time Travel
Time Travel with Statement ID

Question: 77

A resource monitor named MONTHLY_FINANCE_LIMIT has been created and applied to two virtual

warehouses (fin_wh1 and fin_wh2) using the following SQL:

```
ALTER RESOURCE MONITOR MONTHLY_FINANCE_LIMIT SET CREDIT-QUOTA = 1000
    FREQUENCY = MONTHLY
    START_TIMESTAMP = '2022-12-01 08:00 PST'
    NOTIFY_USERS_ON_80_PERCENT DO SUSPEND
    NOTIFY_USERS_ON_100_PERCENT DO SUSPEND_IMMEDIATE;

ALTER WAREHOUSE fin.whl SET RESOURCE-MONITOR = MONTHLY_FINANCE_LIMIT;
ALTER WAREHOUSE fin„wh2 SET RESOURCE.MONITOR = MONTHLY_FINANCE-LIMIT;
```

Given that the combined total of credits consumed by fin_wh1 and fin_wh2 (including cloud services) has reached 800 credits and both warehouses are suspended, what should the ACCOUNTADMIN execute to allow both warehouses to be resumed? (Select TWO).

A. ALTER WAREHOUSE fin_wh1 RESUME;
B. ALTER WAREHOUSE fin_wh2 RESUME;
C. ALTER WAREHOUSE fin_wh1 UNSET RESOURCE_MONITOR MONTHLY_FINANCE_LIMIT;
D. ALTER WAREHOUSE fin_wh2 UNSET RESOURCE_MONITOR MONTHLY_FINANCE_LIMIT;
E. ALTER RESOURCE MONITOR MONTHLY_FINANCE_LIMIT SET CREDIT_QUOTA = 1500;
F. ALTER RESOURCE MONITOR MONTHLY_FINANCE_LIMIT RESET;
G. ALTER WAREHOUSE fin_wh1 UNSET RESOURCE_MONITORS;

Answer: EF

Explanation:

Scenario:

Resource Monitor MONTHLY_FINANCE_LIMIT has a credit quota of 1000.

800 credits have been used and warehouses are already suspended.

According to monitor configuration:

At 80%, warehouses are suspended.

At 100%, warehouses would be suspended immediately.

Warehouses cannot resume until the monitor is reset or the quota is increased.

Q E. SET CREDIT_QUOTA = 1500

Increases the monthly credit limit to 1500.

Since current usage is 800 < 1500, this puts usage below 80%.

This allows resumption of warehouses.

Q F. RESET

sql

CopyEdit

ALTER RESOURCE MONITOR MONTHLY_FINANCE_LIMIT RESET;

Resets usage to zero for the current period.

Allows warehouses to resume immediately — same effect as a fresh cycle.

X Why Other Options Are Incorrect:

A . / B. ALTER WAREHOUSE ... RESUME

Won't work while the resource monitor is actively suspending the warehouses due to limits.

C . / D. UNSET RESOURCE_MONITOR

You can't remove a resource monitor from a warehouse while it is currently suspended by that same monitor.

You must first reset or adjust the monitor.

G . UNSET RESOURCE_MONITORS

Invalid syntax — there's no RESOURCE_MONITORS plural keyword.

SnowPro Administrator Reference:

Resource Monitors Overview

ALTER RESOURCE MONITOR

Best Practices for Controlling Warehouse Credit Usage

Question: 78

An Administrator needs to implement an access control mechanism across an organization. The organization users access sensitive customer data that comes from different regions and needs to be accessible for Analysts who work in these regions. Some Analysts need very specific access control depending on their functional roles in the organization. Following Snowflake recommended practice, how should these requirements be met? (Select TWO).

A. Implement views on top of base tables that exclude or mask sensitive data.
B. Implement row access policies and Dynamic Data Masking policies.
C. Include masking rules as part of data ingestion and transformation pipelines.
D. Use a third-party tool to share the data.
E. Use zero-copy cloning to replicate the database schema and provide access as needed.

Answer: AB

Explanation:

The scenario describes a need for fine-grained access control over sensitive customer data across multiple regions, with functional-role-based access for analysts. Snowflake recommends applying a layered security model that separates raw data from user-facing access and leverages built-in policy features.

Explanation of Correct Answers:

A . Implement views on top of base tables that exclude or mask sensitive data.

Creating secure views allows administrators to abstract sensitive fields or filter out certain rows and columns.

It enables role-based access control by granting specific roles access only to the secure views. Common practice is to restrict access to base tables and give users access to views that enforce business logic and data access rules.

B . Implement row access policies and Dynamic Data Masking policies.

Row Access Policies control access at the row level, determining what data a user can see based on their role or session context.

Dynamic Data Masking allows you to mask sensitive column data (like PII) dynamically based on the accessing role.

Both are central features of Snowflake's fine-grained access control.

Why the other options are incorrect:

C . Include masking rules as part of data ingestion and transformation pipelines.

This is not a Snowflake-recommended best practice for access control.

It hardcodes data access rules into ETL/ELT logic, which reduces flexibility and central control.

Also, it masks the data permanently at ingestion time, rather than dynamically at query time.

D . Use a third-party tool to share the data.

Snowflake supports native Secure Data Sharing, and using a third-party tool is unnecessary and introduces complexity.

It does not address row/column-level access control within Snowflake itself.

E . Use zero-copy cloning to replicate the database schema and provide access as needed. Zero-copy cloning is ideal for testing, development, and backup purposes, not for controlling access. It duplicates metadata but doesn't provide a mechanism for fine-grained, real-time access control.

SnowPro Administrator Reference:

Row Access Policies Overview

Dynamic Data Masking Overview

Access Control Best Practices

Using Secure Views for Access Control