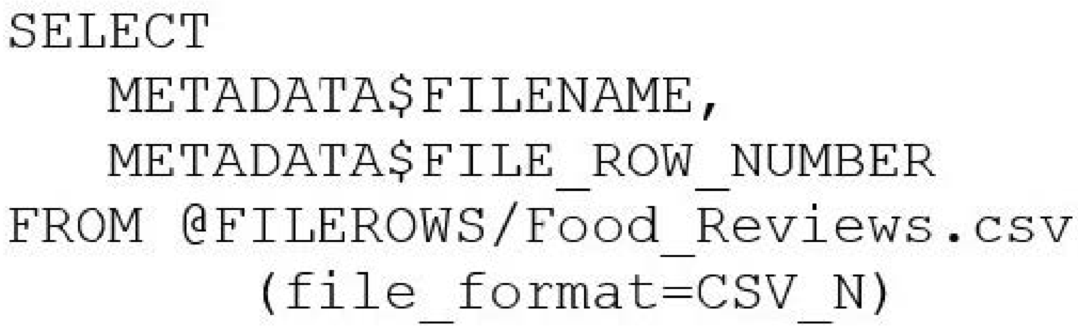
ARA-C01$^{\text{Q\&As}}$

SnowPro Advanced: Architect Certification Exam

Pass Snowflake ARA-C01 Exam with 100% Guarantee

QUESTION 1

An Architect runs the following SQL query:



```
SELECT
    METADATA$FILENAME,
    METADATA$FILE_ROW_NUMBER
FROM @FILEROWS/Food_Reviews.csv
          (file_format=CSV_N)
```

How can this query be interpreted?

A. FILEROWS is a stage. FILE_ROW_NUMBER is line number in file.

B. FILEROWS is the table. FILE_ROW_NUMBER is the line number in the table.

C. FILEROWS is a file. FILE_ROW_NUMBER is the file format location.

D. FILERONS is the file format location. FILE_ROW_NUMBER is a stage.

Correct Answer: A

A stage is a named location in Snowflake that can store files for data loading and unloading. A stage can be internal or external, depending on where the files are stored.

The query in the question uses the LIST function to list the files in a stage named FILEROWS. The function returns a table with various columns, including FILE_ROW_NUMBER, which is the line number of the file in the stage. Therefore, the

query can be interpreted as listing the files in a stage named FILEROWS and showing the line number of each file in the stage.

References:

: Stages

: LIST Function

QUESTION 2

Which of the following are characteristics of how row access policies can be applied to external tables? (Choose three.)

A. An external table can be created with a row access policy, and the policy can be applied to the VALUE column.

B. A row access policy can be applied to the VALUE column of an existing external table.

C. A row access policy cannot be directly added to a virtual column of an external table.

D. External tables are supported as mapping tables in a row access policy.

E. While cloning a database, both the row access policy and the external table will be cloned.

F. A row access policy cannot be applied to a view created on top of an external table.

Correct Answer: ABC

Explanation: These three statements are true according to the Snowflake documentation and the web search results. A row access policy is a feature that allows filtering rows based on user-defined conditions. A row access policy can be applied to an external table, which is a table that reads data from external files in a stage. However, there are some limitations and considerations for using row access policies with external tables. An external table can be created with a row access policy by using the WITH ROW ACCESS POLICY clause in the CREATE EXTERNAL TABLE statement. The policy can be applied to the VALUE column, which is the column that contains the raw data from the external files in a VARIANT data type1. A row access policy can also be applied to the VALUE column of an existing external table by using the ALTER TABLE statement with the SET ROW ACCESS POLICY clause2. A row access policy cannot be directly added to a virtual column of an external table. A virtual column is a column that is derived from the VALUE column using an expression. To apply a row access policy to a virtual column, the policy must be applied to the VALUE column and the expression must be repeated in the policy definition3. External tables are not supported as mapping tables in a row access policy. A mapping table is a table that is used to determine the access rights of users or roles based on some criteria. Snowflake does not support using an external table as a mapping table because it may cause performance issues or errors4. While cloning a database, Snowflake clones the row access policy, but not the external table. Therefore, the policy in the cloned database refers to a table that is not present in the cloned database. To avoid this issue, the external table must be manually cloned or recreated in the cloned database4. A row access policy can be applied to a view created on top of an external table. The policy can be applied to the view itself or to the underlying external table. However, if the policy is applied to the view, the view must be a secure view, which is a view that hides the underlying data and the view definition from unauthorized users5. References: CREATE EXTERNAL TABLE | Snowflake Documentation ALTER EXTERNAL TABLE | Snowflake Documentation Understanding Row Access Policies | Snowflake Documentation Snowflake Data Governance: Row Access Policy Overview Secure Views | Snowflake Documentation

QUESTION 3

What are purposes for creating a storage integration? (Choose three.)

A. Control access to Snowflake data using a master encryption key that is maintained in the cloud provider\\'s key management service.

B. Store a generated identity and access management (IAM) entity for an external cloud provider regardless of the cloud provider that hosts the Snowflake account.

C. Support multiple external stages using one single Snowflake object.

D. Avoid supplying credentials when creating a stage or when loading or unloading data.

E. Create private VPC endpoints that allow direct, secure connectivity between VPCs without traversing the public internet.

F. Manage credentials from multiple cloud providers in one single Snowflake object.

Correct Answer: BCD

A storage integration is a Snowflake object that stores a generated identity and access management (IAM) entity for an external cloud provider, such as Amazon S3, Google Cloud Storage, or Microsoft Azure Blob Storage. This integration allows Snowflake to read data from and write data to an external storage location referenced in an external stage1. One purpose of creating a storage integration is to support multiple external stages using one single Snowflake object. An integration can list buckets (and optional paths) that limitthe locations users can specify when creating external stages that use the integration. Note that many external stage objects can reference different buckets and paths and use the same storage integration for authentication1. Therefore, option C is correct. Another purpose of creating a storage integration is to avoid supplying credentials when creating a stage or when loading or unloading data. Integrations are named, first-class Snowflake objects that avoid the need for passing explicit cloud provider credentials such as secret keys or access tokens. Integration objects store an IAM user ID, and an administrator in your organization grants the IAM user permissions in the cloud provider account1. Therefore, option D is correct. A third purpose of creating a storage integration is to store a generated IAM entity for an external cloud provider regardless of the cloud provider that hosts the Snowflake account. For example, you can create a storage integration for Amazon S3 even if your Snowflake account is hosted on Azure or Google Cloud Platform. This allows you to access data across different cloud platforms using Snowflake1. Therefore, option B is correct. Option A is incorrect, because creating a storage integration does not control access to Snowflake data using a master encryption key. Snowflake encrypts all data using a hierarchical key model, and the master encryption key is managed by Snowflake or by the customer using a cloud provider\\'s key management service. This is independent of the storage integration feature2. Option E is incorrect, because creating a storage integration does not create private VPC endpoints. Private VPC endpoints are a network configuration option that allow direct, secure connectivity between VPCs without traversing the public internet. This is also independent of the storage integration feature3. Option F is incorrect, because creating a storage integration does not manage credentials from multiple cloud providers in one single Snowflake object. A storage integration is specific to one cloud provider, and you need to create separate integrations for each cloud provider you want to access4. References: : Encryption and Decryption : Private Link for Snowflake : CREATE STORAGE INTEGRATION : Option 1: Configuring a Snowflake Storage Integration to Access Amazon S3

QUESTION 4

A DevOps team has a requirement for recovery of staging tables used in a complex set of data pipelines. The staging tables are all located in the same staging schema. One of the requirements is to have online recovery of data on a rolling 7day basis.

After setting up the DATA_RETENTION_TIME_IN_DAYS at the database level, certain tables remain unrecoverable past 1 day.

What would cause this to occur? (Choose two.)

A. The staging schema has not been setup for MANAGED ACCESS.

B. The DATA_RETENTION_TIME_IN_DAYS for the staging schema has been set to 1 day.

C. The tables exceed the 1 TB limit for data recovery.

D. The staging tables are of the TRANSIENT type.

E. The DevOps role should be granted ALLOW_RECOVERY privilege on the staging schema.

Correct Answer: BD

The DATA_RETENTION_TIME_IN_DAYS parameter controls the Time Travel retention period for an object (database, schema, or table) in Snowflake. This parameter specifies the number of days for which historical data is preserved and can be accessed using Time Travel operations (SELECT, CREATE ... CLONE, UNDROP)1. The requirement for recovery of staging tables on a rolling 7-day basis means that the DATA_RETENTION_TIME_IN_DAYS parameter should be set to 7 at the database level. However, this parameter can be overridden at the lower levels (schema or table) if they have a different value1. Therefore, one possible cause for certain tables to remain unrecoverable past 1 day is that the DATA_RETENTION_TIME_IN_DAYS for the staging schema has been set to 1 day. This would override the database level setting and limit the Time Travel retention period for all the tables in the schema to 1 day. To fix this, the parameter should be unset or set to 7 at the schema level1. Therefore, option B is correct. Another possible cause for certain tables to remain unrecoverable past 1 day is that the staging tables are of the TRANSIENT type. Transient tables are tables that do not have a Fail-safe period and can have a Time Travel retention period of either 0 or 1 day. Transient tables are suitable for temporary or intermediate data that can be easily reproduced or replicated2. To fix this, the tables should be created as permanent tables, which can have a Time Travel retention period of up to 90 days1. Therefore, option D is correct. Option A is incorrect because the MANAGED ACCESS feature is not related to the data recovery requirement. MANAGED ACCESS is a feature that allows granting access privileges to objects without explicitly granting the privileges to roles. It does not affect the Time Travel retention period or the data availability3. Option C is incorrect because there is no 1 TB limit for data recovery in Snowflake. The data storage size does not affect the Time Travel retention period or the data availability4. Option E is incorrect because there is no ALLOW_RECOVERY privilege in Snowflake. The privilege required to perform Time Travel operations is SELECT, which allows querying historical data in tables5. References: : Understanding and Using Time Travel : Transient Tables : Managed Access : Understanding Storage Cost : Table Privileges

QUESTION 5

What Snowflake features should be leveraged when modeling using Data Vault? (Choose two.)

A. Snowflake\\'s support of multi-table inserts into the data model\\'s Data Vault tables

B. Data needs to be pre-partitioned to obtain a superior data access performance

C. Scaling up the virtual warehouses will support parallel processing of new source loads

D. Snowflake\\'s ability to hash keys so that hash key joins can run faster than integer joins

Correct Answer: AC

Explanation: These two features are relevant for modeling using Data Vault on Snowflake. Data Vault is a data modeling approach that organizes data into hubs, links, and satellites. Data Vault is designed to enable high scalability, flexibility, and performance for data integration and analytics. Snowflake is a cloud data platform that supports various data modeling techniques, including Data Vault. Snowflake provides some features that can enhance the Data Vault modeling, such as: Snowflake\\'s support of multi-table inserts into the data model\\'s Data Vault tables. Multi-table inserts (MTI) are a feature that allows inserting data from a single query into multiple tables in a single DML statement. MTI can improve the performance and efficiency of loading datainto Data Vault tables, especially for real-time or near-real-time data integration. MTI can also reduce the complexity and maintenance of the loading code, as well as the data duplication and latency12. Scaling up the virtual warehouses will support parallel processing of new source loads. Virtual warehouses are a feature that allows provisioning compute resources on demand for data processing. Virtual warehouses can be scaled up or down by changing the size of the warehouse, which determines the number of servers in the warehouse. Scaling up the virtual warehouses can improve the performance and concurrency of processing new source loads into Data Vault tables, especially for large or complex data sets. Scaling up the virtual warehouses can also leverage the parallelism and distribution of Snowflake\\'s architecture, which can optimize the data loading and querying34. References: Snowflake Documentation: Multi-table Inserts Snowflake Blog: Tips for Optimizing the Data Vault Architecture on Snowflake Snowflake Documentation: Virtual Warehouses Snowflake Blog: Building a Real-Time Data Vault in Snowflake