ARA-C01 Q&As

SnowPro Advanced: Architect Certification Exam

Pass Snowflake ARA-C01 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.pass4itsure.com/ara-c01.html

100% Passing Guarantee
100% Money Back Assurance

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

QUESTION 1

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 2

A media company needs a data pipeline that will ingest customer review data into a Snowflake table, and apply some transformations. The company also needs to use Amazon Comprehend to do sentiment analysis and make the deidentified final data set available publicly for advertising companies who use different cloud providers in different regions.

The data pipeline needs to run continuously ang efficiently as new records arrive in the object storage leveraging event notifications. Also, the operational complexity, maintenance of the infrastructure, including platform upgrades and security, and the development effort should be minimal.

Which design will meet these requirements?

A. Ingest the data using COPY INTO and use streams and tasks to orchestrate transformations. Export the data into Amazon S3 to do model inference with Amazon Comprehend and ingest the data back into a Snowflake table. Then create a listing in the Snowflake Marketplace to make the data available to other companies.

B. Ingest the data using Snowpipe and use streams and tasks to orchestrate transformations. Create an external function to do model inference with Amazon Comprehend and write the final records to a Snowflake table. Then create a listing in the Snowflake Marketplace to make the data available to other companies.

C. Ingest the data into Snowflake using Amazon EMR and PySpark using the Snowflake Spark connector. Apply transformations using another Spark job. Develop a python program to do model inference by leveraging the Amazon Comprehend text analysis API. Then write the results to a Snowflake table and create a listing in the Snowflake Marketplace to make the data available to other companies.

D. Ingest the data using Snowpipe and use streams and tasks to orchestrate transformations. Export the data into Amazon S3 to do model inference with Amazon Comprehend and ingest the data back into a Snowflake table. Then create a listing in the Snowflake Marketplace to make the data available to other companies.

Correct Answer: B

Explanation: This design meets all the requirements for the data pipeline. Snowpipe is a feature that enables continuous data loading into Snowflake from object storage using event notifications. It is efficient, scalable, and serverless,

meaning it does not require any infrastructure or maintenance from the user. Streams and tasks are features that enable automated data pipelines within Snowflake, using change data capture and scheduled execution. They are also efficient,

scalable, and serverless, and they simplify the data transformation process. External functions are functions that can invoke external services or APIs from within Snowflake. They can be used to integrate with Amazon Comprehend and

perform sentiment analysis on the data. The results can be written back to a Snowflake table using standard SQL commands. Snowflake Marketplace is a platform that allows data providers to share data with data consumers across different

accounts, regions, and cloud platforms. It is a secure and easy way to make data publicly available to other companies.

References:

Snowpipe Overview | Snowflake Documentation

Introduction to Data Pipelines | Snowflake Documentation External Functions Overview | Snowflake Documentation Snowflake Data Marketplace Overview | Snowflake Documentation

QUESTION 3

A company is storing large numbers of small JSON files (ranging from 1-4 bytes) that are received from IoT devices and sent to a cloud provider. In any given hour, 100,000 files are added to the cloud provider.

What is the MOST cost-effective way to bring this data into a Snowflake table?

A. An external table

B. A pipe

C. A stream

D. A copy command at regular intervals

Correct Answer: B

A pipe is a Snowflake object that continuously loads data from files in a stage (internal or external) into a table. A pipe can be configured to use auto-ingest, which means that Snowflake automatically detects new or modified files in the stage and loads them into the table without any manual intervention1. A pipe is the most cost-effective way to bring large numbers of small JSON files into a Snowflake table, because it minimizes the number of COPY commands executed and the number of micro-partitions created. A pipe can use file aggregation, which means that it can combine multiple small files into a single larger file before loading them into the table. This reduces the load time and the storage cost of the data2. An external table is a Snowflake object that references data files stored in an external location, such as Amazon S3, Google Cloud Storage, or Microsoft Azure Blob Storage. An external table does not store the data in Snowflake, but only provides a view of the data for querying. An external table is not a cost-effective way to bring data into a Snowflake table, because it does not support file aggregation, and it requires additional network bandwidth and compute resources to

query the external data3.

A stream is a Snowflake object that records the history of changes (inserts, updates, and deletes) made to a table. A stream can be used to consume the changes from a table and apply them to another table or a task. A stream is not a way

to bring data into a Snowflake table, but a way to process the data after it is loaded into a table4.

A copy command is a Snowflake command that loads data from files in a stage into a table. A copy command can be executed manually or scheduled using a task. A copy command is not a cost-effective way to bring large numbers of small

JSON files into a Snowflake table, because it does not support file aggregation, and it may create many micro-partitions that increase the storage cost of the data5.

References: : Pipes : Loading Data Using Snowpipe : External Tables : Streams : COPY INTO

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

A company\\'s client application supports multiple authentication methods, and is using Okta.

What is the best practice recommendation for the order of priority when applications authenticate to Snowflake?

A. 1) OAuth (either Snowflake OAuth or External OAuth) 2) External browser 3) Okta native authentication 4) Key Pair Authentication, mostly used for service account users

5) Password

B. 1) External browser, SSO 2) Key Pair Authentication, mostly used for development environment users 3) Okta native authentication 4) OAuth (ether Snowflake OAuth or External OAuth) 5) Password

C. 1) Okta native authentication 2) Key Pair Authentication, mostly used for production environment users 3) Password 4) OAuth (either Snowflake OAuth or External OAuth) 5) External browser, SSO

D. 1) Password 2) Key Pair Authentication, mostly used for production environment users 3) Okta native authentication 4) OAuth (either Snowflake OAuth or External OAuth) 5) External browser, SSO

Correct Answer: A

This is the best practice recommendation for the order of priority when applications authenticate to Snowflake, according to the Snowflake documentation and the web search results. Authentication is the process of verifying the identity of a user or application that connects to Snowflake. Snowflake supports multiple authentication methods, each with different advantages and disadvantages. The recommended order of priority is based on the following factors: Security: The authentication method should provide a high level of security and protection against unauthorized access or data breaches. The authentication method should also support multi-factor authentication (MFA) or single sign-on (SSO) for additional security. Convenience: The authentication method should provide a smooth and easy user experience, without requiring complex or manual steps. The authentication method should also support seamless integration with external identity providers or applications. Flexibility: The authentication method should provide a range of options and features to suit different use cases and scenarios. The authentication method should also support customization and configuration to meet specific requirements. Based on these factors, the recommended order of

priority is: OAuth (either Snowflake OAuth or External OAuth): OAuth is an open standard for authorization that allows applications to access Snowflake resources on behalf of a user, without exposing the user\\'s credentials. OAuth provides a high level of security, convenience, and flexibility, as it supports MFA, SSO, token-based authentication, and various grant types and scopes. OAuth can be implemented using either Snowflake OAuth or External OAuth, depending on the identity provider and the application12. External browser: External browser is an authentication method that allows users to log in to Snowflake using a web browser and an external identity provider, such as Okta, Azure AD, or Ping Identity. External browser provides a high level of security and convenience, as it supports MFA, SSO, and federated authentication. External browser also provides a consistent user interface and experience across different platforms and devices34. Okta native authentication: Okta native authentication is an authentication method that allows users to log in to Snowflake using Okta as the identity provider, without using a web browser. Okta native authentication provides a high level of security and convenience, as it supports MFA, SSO, and federated authentication. Okta native authentication also provides a native user interface and experience for Okta users, and supports various Okta features, such as password policies and user management56. Key Pair Authentication: Key Pair Authentication is an authentication method that allows users to log in to Snowflake using a public-private key pair, without using a password. Key Pair Authentication provides a high level of security, as it relies on asymmetric encryption and digital signatures. Key Pair Authentication also provides a flexible and customizable authentication option, as it supports various key formats, algorithms, and expiration times. Key Pair Authentication is mostly used for service account users, such as applications or scripts that connect to Snowflake programmatically7 . Password: Password is the simplest and most basic authentication method that allows users to log in to Snowflake using a username and password. Password provides a low level of security, as it relies on symmetric encryption and is vulnerable to brute force attacks or phishing. Password also provides a low level of convenience and flexibility, as it requires manual input and management, and does not support MFA or SSO. Password is the least recommended authentication method, and should be used only as a last resort or for testing purposes . References: Snowflake Documentation: Snowflake OAuth Snowflake Documentation: External OAuth Snowflake Documentation: External Browser Authentication Snowflake Blog: How to Use External Browser Authentication with Snowflake Snowflake Documentation: Okta Native Authentication Snowflake Blog: How to Use Okta Native Authentication with Snowflake Snowflake Documentation: Key Pair Authentication [Snowflake Blog: How to Use Key Pair Authentication with Snowflake] [Snowflake Documentation: Password Authentication] [Snowflake Blog: How to Use Password Authentication with Snowflake]