Snowflake ARA-R01 Practice Questions

Shared by Larsen on 19-08-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

An Architect uses COPY INTO with the ON_ERROR=SKIP_FILE option to bulk load CSV files into a table called TABLEA, using its table stage. One file named file5.csv fails to load. The Architect fixes the file and re-loads it to the stage with the exact same file name it had previously.

Which commands should the Architect use to load only file5.csv file from the stage? (Choose two.)

Options:

A- COPY INTO tablea FROM @%tablea RETURN_FAILED_ONLY = TRUE;
B- COPY INTO tablea FROM @%tablea;
C- COPY INTO tablea FROM @%tablea FILES = ('file5.csv');
D- COPY INTO tablea FROM @%tablea FORCE = TRUE;
E- COPY INTO tablea FROM @%tablea NEW_FILES_ONLY = TRUE;
F- COPY INTO tablea FROM @%tablea MERGE = TRUE;

Answer:

B, C

Explanation:

Option A (RETURN_FAILED_ONLY)will only load files that previously failed to load.Since file5.csv already exists in the stage with the same name,it will not be considered a new file and will not be loaded.

Option D (FORCE)will overwrite any existing data in the table.This is not desired as we only want to load the data from file5.csv.

Option E (NEW_FILES_ONLY)will only load files that have been added to the stage since the last COPY command.This will not work because file5.csv was already in the stage before it was fixed.

Option F (MERGE)is used to merge data from a stage into an existing table,creating new rows for any data not already present.This is not needed in this case as we simply want to load the data from file5.csv.

Therefore, the architect can use either COPY INTO tablea FROM @%tablea or COPY INTO tablea FROM @%tablea FILES = ('file5.csv') to load only file5.csv from the stage. Both options will load the data from the specified file without overwriting any existing data or requiring additional configuration

Question 2

Question Type: MultipleChoice

A retail company has over 3000 stores all using the same Point of Sale (POS) system. The company wants to deliver near real-time sales results to category managers. The stores operate in a variety of time zones and exhibit a dynamic range of transactions each minute, with some stores having higher sales volumes than others.

Sales results are provided in a uniform fashion using data engineered fields that will be calculated in a complex data pipeline. Calculations include exceptions, aggregations, and scoring using external functions interfaced to scoring algorithms. The source data for aggregations has over 100M rows.

Every minute, the POS sends all sales transactions files to a cloud storage location with a naming convention that includes store numbers and timestamps to identify the set of transactions contained in the files. The files are typically less than 10MB in size.

How can the near real-time results be provided to the category managers? (Select TWO).

Options:

A- All files should be concatenated before ingestion into Snowflake to avoid micro-ingestion.
B- A Snowpipe should be created and configured with AUTO_INGEST = true. A stream should be created to process INSERTS into a single target table using the stream metadata to inform the store number and timestamps.
C- A stream should be created to accumulate the near real-time data and a task should be created that runs at a frequency that matches the real-time analytics needs.
D- An external scheduler should examine the contents of the cloud storage location and issue SnowSQL commands to process the data at a frequency that matches the real-time analytics needs.
E- The copy into command with a task scheduled to run every second should be used to achieve the near-real time requirement.

Answer:

B, C

Explanation:

To provide near real-time sales results to category managers, the Architect can use the following steps:

Create an external stage that references the cloud storage location where the POS sends the

sales transactions files.The external stage should use the file format and encryption settings that match the source files2

Create a Snowpipe that loads the files from the external stage into a target table in Snowflake. The Snowpipe should be configured with AUTO_INGEST = true, which means that it will automatically detect and ingest new files as they arrive in the external stage.The Snowpipe should also use a copy option to purge the files from the external stage after loading, to avoid duplicate ingestion3

Create a stream on the target table that captures the INSERTS made by the Snowpipe. The stream should include the metadata columns that provide information about the file name, path, size, and last modified time.The stream should also have a retention period that matches the real-time analytics needs4

Create a task that runs a query on the stream to process the near real-time data. The query should use the stream metadata to extract the store number and timestamps from the file name and path, and perform the calculations for exceptions, aggregations, and scoring using external functions. The query should also output the results to another table or view that can be accessed by the category managers. The task should be scheduled to run at a frequency that matches the real-time analytics needs, such as every minute or every 5 minutes.

The other options are not optimal or feasible for providing near real-time results:

All files should be concatenated before ingestion into Snowflake to avoid micro-ingestion. This option is not recommended because it would introduce additional latency and complexity in the data pipeline. Concatenating files would require an external process or service that monitors the cloud storage location and performs the file merging operation. This would delay the ingestion of new files into Snowflake and increase the risk of data loss or corruption. Moreover, concatenating files would not avoid micro-ingestion, as Snowpipe would still ingest each concatenated file as a separate load.

An external scheduler should examine the contents of the cloud storage location and issue SnowSQL commands to process the data at a frequency that matches the real-time analytics needs. This option is not necessary because Snowpipe can automatically ingest new files from the external stage without requiring an external trigger or scheduler. Using an external scheduler would add more overhead and dependency to the data pipeline, and it would not guarantee near real-time ingestion, as it would depend on the polling interval and the availability of the external scheduler.

The copy into command with a task scheduled to run every second should be used to achieve the near-real time requirement. This option is not feasible because tasks cannot be scheduled to run every second in Snowflake. The minimum interval for tasks is one minute, and even that is not guaranteed, as tasks are subject to scheduling delays and concurrency limits. Moreover, using the copy into command with a task would not leverage the benefits of Snowpipe, such as automatic file detection, load balancing, and micro-partition optimization.Reference:

1: SnowPro Advanced: Architect | Study Guide

2: Snowflake Documentation | Creating Stages

3: Snowflake Documentation | Loading Data Using Snowpipe

4: Snowflake Documentation | Using Streams and Tasks for ELT

: Snowflake Documentation | Creating Tasks

: Snowflake Documentation | Best Practices for Loading Data

: Snowflake Documentation | Using the Snowpipe REST API

: Snowflake Documentation | Scheduling Tasks

:SnowPro Advanced: Architect | Study Guide

:Creating Stages

:Loading Data Using Snowpipe

:Using Streams and Tasks for ELT

: [Creating Tasks]

: [Best Practices for Loading Data]

: [Using the Snowpipe REST API]

: [Scheduling Tasks]

Question 3

Question Type: MultipleChoice

A retailer's enterprise data organization is exploring the use of Data Vault 2.0 to model its data lake solution. A Snowflake Architect has been asked to provide recommendations for using Data Vault 2.0 on Snowflake.

What should the Architect tell the data organization? (Select TWO).

Options:

A- Change data capture can be performed using the Data Vault 2.0 HASH_DIFF concept.
B- Change data capture can be performed using the Data Vault 2.0 HASH_DELTA concept.
C- Using the multi-table insert feature in Snowflake, multiple Point-in-Time (PIT) tables can be loaded in parallel from a single join query from the data vault.
D- Using the multi-table insert feature, multiple Point-in-Time (PIT) tables can be loaded sequentially from a single join query from the data vault.
E- There are performance challenges when using Snowflake to load multiple Point-in-Time (PIT) tables in parallel from a single join query from the data vault.

Answer:

A, C

Explanation:

Data Vault 2.0 on Snowflake supports the HASH_DIFF concept for change data capture, which is a method to detect changes in the data by comparing the hash values of the records. Additionally, Snowflake's multi-table insert feature allows for the loading of multiple PIT tables in parallel from a single join query, which can significantly streamline the data loading process and improve performance1.

Reference =

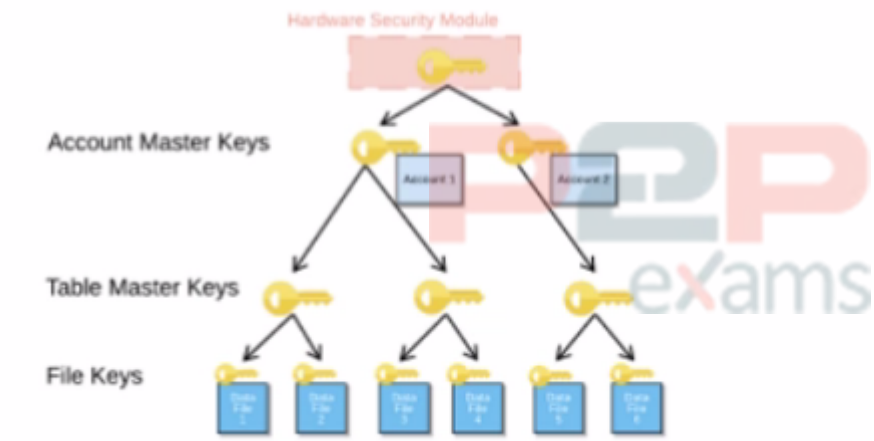
* Snowflake's documentation on multi-table inserts1

* Blog post on optimizing Data Vault architecture on Snowflake2

Question 4

Question Type: MultipleChoice

When activating Tri-Secret Secure in a hierarchical encryption model in a Snowflake account, at what level is the customer-managed key used?



Options:

A- At the root level (HSM)
B- At the account level (AMK)
C- At the table level (TMK)
D- At the micro-partition level

Answer:

B

Explanation:

Tri-Secret Secure is a feature that allows customers to use their own key, called the customer-managed key (CMK), in addition to the Snowflake-managed key, to create a composite master key that encrypts the data in Snowflake. The composite master key is also known as the account master key (AMK), as it is unique for each account and encrypts the table master keys (TMKs) that encrypt the file keys that encrypt the data files. The customer-managed key is used at the account level, not at the root level, the table level, or the micro-partition level.The root level is protected by a hardware security module (HSM), the table level is protected by the TMKs, and the micro-partition level is protected by the file keys12.Reference:

Understanding Encryption Key Management in Snowflake

Tri-Secret Secure FAQ for Snowflake on AWS

Question 5

Question Type: MultipleChoice

An Architect has a design where files arrive every 10 minutes and are loaded into a primary database table using Snowpipe. A secondary database is refreshed every hour with the latest data from the primary database.

Based on this scenario, what Time Travel query options are available on the secondary database?

Options:

A- A query using Time Travel in the secondary database is available for every hourly table version within the retention window.
B- A query using Time Travel in the secondary database is available for every hourly table version within and outside the retention window.
C- Using Time Travel, secondary database users can query every iterative version within each hour (the individual Snowpipe loads) in the retention window.
D- Using Time Travel, secondary database users can query every iterative version within each hour (the individual Snowpipe loads) and outside the retention window.

Answer:

A

Explanation:

Snowflake's Time Travel feature allows users to query historical data within a defined retention period. In the given scenario, since the secondary database is refreshed every hour, Time Travel can be used to query each hourly version of the table as long as it falls within the retention window. This does not include individual Snowpipe loads within each hour unless they coincide with the hourly refresh.

Question 6

Question Type: MultipleChoice

What is a valid object hierarchy when building a Snowflake environment?

Options:

A- Account --> Database --> Schema --> Warehouse
B- Organization --> Account --> Database --> Schema --> Stage
C- Account --> Schema > Table --> Stage
D- Organization --> Account --> Stage --> Table --> View

Answer:

B

Explanation:

This is the valid object hierarchy when building a Snowflake environment, according to the Snowflake documentation and the web search results. Snowflake is a cloud data platform that supports various types of objects, such as databases, schemas, tables, views, stages, warehouses, and more. These objects are organized in a hierarchical structure, as follows:

Organization: An organization is the top-level entity that represents a group of Snowflake accounts that are related by business needs or ownership.An organization can have one or more accounts, and can enable features such as cross-account data sharing, billing and usage reporting, and single sign-on across accounts12.

Account: An account is the primary entity that represents a Snowflake customer. An account

can have one or more databases, schemas, stages, warehouses, and other objects. An account can also have one or more users, roles, and security integrations.An account is associated with a specific cloud platform, region, and Snowflake edition34.

Database: A database is a logical grouping of schemas. A database can have one or more schemas, and can store structured, semi-structured, or unstructured data.A database can also have properties such as retention time, encryption, and ownership56.

Schema: A schema is a logical grouping of tables, views, stages, and other objects. A schema can have one or more objects, and can define the namespace and access control for the objects. A schema can also have properties such as ownership and default warehouse .

Stage: A stage is a named location that references the files in external or internal storage. A stage can be used to load data into Snowflake tables using the COPY INTO command, or to unload data from Snowflake tables using the COPY INTO LOCATION command. A stage can be created at the account, database, or schema level, and can have properties such as file format, encryption, and credentials .

The other options listed are not valid object hierarchies, because they either omit or misplace some objects in the structure. For example, option A omits the organization level and places the warehouse under the schema level, which is incorrect. Option C omits the organization, account, and stage levels, and places the table under the schema level, which is incorrect. Option D omits the database level and places the stage and table under the account level, which is incorrect.

Snowflake Documentation: Organizations

Snowflake Blog: Introducing Organizations in Snowflake

Snowflake Documentation: Accounts

Snowflake Blog: Understanding Snowflake Account Structures

Snowflake Documentation: Databases

Snowflake Blog: How to Create a Database in Snowflake

[Snowflake Documentation: Schemas]

[Snowflake Blog: How to Create a Schema in Snowflake]

[Snowflake Documentation: Stages]

[Snowflake Blog: How to Use Stages in Snowflake]

Question 7

Question Type: MultipleChoice

A company has several sites in different regions from which the company wants to ingest data.

Which of the following will enable this type of data ingestion?

Options:

A- The company must have a Snowflake account in each cloud region to be able to ingest data to that account.
B- The company must replicate data between Snowflake accounts.
C- The company should provision a reader account to each site and ingest the data through the reader accounts.
D- The company should use a storage integration for the external stage.

Answer:

D

Explanation:

This is the correct answer because it allows the company to ingest data from different regions using a storage integration for the external stage. A storage integration is a feature that enables secure and easy access to files in external cloud storage from Snowflake. A storage integration can be used to create an external stage, which is a named location that references the files in the external storage. An external stage can be used to load data into Snowflake tables using the COPY INTO command, or to unload data from Snowflake tables using the COPY INTO LOCATION command.A storage integration can support multiple regions and cloud platforms, as long as the external storage service is compatible with Snowflake12.

Snowflake Documentation: Storage Integrations

Snowflake Documentation: External Stages

Question 8

Question Type: MultipleChoice

Which Snowflake data modeling approach is designed for BI queries?

Options:

A- 3 NF
B- Star schema
C- Data Vault
D- Snowflake schema

Answer:

B

Explanation:

In the context of business intelligence (BI) queries, which are typically focused on data analysis and reporting, the star schema is the most suitable data modeling approach.

Option B: Star Schema - The star schema is a type of relational database schema that is widely used for developing data warehouses and data marts for BI purposes. It consists of a central fact table surrounded by dimension tables. The fact table contains the core data metrics, and the dimension tables contain descriptive attributes related to the fact data. The simplicity of the star schema allows for efficient querying and aggregation, which are common operations in BI reporting.

Question 9

Question Type: MultipleChoice

A Data Engineer is designing a near real-time ingestion pipeline for a retail company to ingest event logs into Snowflake to derive insights. A Snowflake Architect is asked to define security best practices to configure access control privileges for the data load for auto-ingest to Snowpipe.

What are the MINIMUM object privileges required for the Snowpipe user to execute Snowpipe?

Options:

A- OWNERSHIP on the named pipe, USAGE on the named stage, target database, and schema, and INSERT and SELECT on the target table
B- OWNERSHIP on the named pipe, USAGE and READ on the named stage, USAGE on the target database and schema, and INSERT end SELECT on the target table
C- CREATE on the named pipe, USAGE and READ on the named stage, USAGE on the target database and schema, and INSERT end SELECT on the target table
D- USAGE on the named pipe, named stage, target database, and schema, and INSERT and

SELECT on the target table

Answer:

B

Explanation:

According to the SnowPro Advanced: Architect documents and learning resources, the minimum object privileges required for the Snowpipe user to execute Snowpipe are:

OWNERSHIP on the named pipe.This privilege allows the Snowpipe user to create, modify, and drop the pipe object that defines the COPY statement for loading data from the stage to the table1.

USAGE and READ on the named stage.These privileges allow the Snowpipe user to access and read the data files from the stage that are loaded by Snowpipe2.

USAGE on the target database and schema.These privileges allow the Snowpipe user to access the database and schema that contain the target table3.

INSERT and SELECT on the target table.These privileges allow the Snowpipe user to insert data into the table and select data from the table4.

The other options are incorrect because they do not specify the minimum object privileges required for the Snowpipe user to execute Snowpipe. Option A is incorrect because it does not include the READ privilege on the named stage, which is required for the Snowpipe user to read the data files from the stage. Option C is incorrect because it does not include the OWNERSHIP privilege on the named pipe, which is required for the Snowpipe user to create, modify, and drop the pipe object. Option D is incorrect because it does not include the OWNERSHIP privilege on the named pipe or the READ privilege on the named stage, which are both required for the Snowpipe user to execute Snowpipe.Reference:CREATE PIPE | Snowflake Documentation,CREATE STAGE | Snowflake Documentation,CREATE DATABASE | Snowflake Documentation,CREATE TABLE | Snowflake Documentation

Question 10

Question Type: MultipleChoice

A company is trying to Ingest 10 TB of CSV data into a Snowflake table using Snowpipe as part of Its migration from a legacy database platform. The records need to be ingested in the MOST performant and cost-effective way.

How can these requirements be met?

Options:

A- Use ON_ERROR = continue in the copy into command.
B- Use purge = TRUE in the copy into command.
C- Use FURGE = FALSE in the copy into command.
D- Use on error = SKIP_FILE in the copy into command.

Answer:

D

Explanation:

For ingesting a large volume of CSV data into Snowflake using Snowpipe, especially for a substantial amount like 10 TB, the on error = SKIP_FILE option in the COPY INTO command can be highly effective. This approach allows Snowpipe to skip over files that cause errors during the ingestion process, thereby not halting or significantly slowing down the overall data load. It helps in maintaining performance and cost-effectiveness by avoiding the reprocessing of problematic files and continuing with the ingestion of other data.