Snowflake ARA-R01 Mock Exam

Shared by Dejesus on 17-06-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

An Architect is designing a pipeline to stream event data into Snowflake using the Snowflake Kafka connector. The Architect's highest priority is to configure the connector to stream data in the MOST cost-effective manner.

Which of the following is recommended for optimizing the cost associated with the Snowflake Kafka connector?

Options:

A- Utilize a higher Buffer.flush.time in the connector configuration.
B- Utilize a higher Buffer.size.bytes in the connector configuration.
C- Utilize a lower Buffer.size.bytes in the connector configuration.
D- Utilize a lower Buffer.count.records in the connector configuration.

Answer:

A

Explanation:

The minimum value supported for the buffer.flush.time property is 1 (in seconds). For higher average data flow rates, we suggest that you decrease the default value for improved latency. If cost is a greater concern than latency, you could increase the buffer flush time. Be careful to flush the Kafka memory buffer before it becomes full to avoid out of memory exceptions.
https://docs.snowflake.com/en/user-guide/data-load-snowpipe-streaming-kafka

Question 2

Question Type: MultipleChoice

At which object type level can the APPLY MASKING POLICY, APPLY ROW ACCESS POLICY and APPLY SESSION POLICY privileges be granted?

Options:

A- Global
B- Database
C- Schema
D- Table

Answer:

A

Explanation:

The object type level at which the APPLY MASKING POLICY, APPLY ROW ACCESS POLICY and APPLY SESSION POLICY privileges can be granted is global. These are account-level privileges that control who can apply or unset these policies on objects such as columns, tables, views, accounts, or users. These privileges are granted to the ACCOUNTADMIN role by default, and can be granted to other roles as needed. The other options are incorrect because they are not the object type level at which these privileges can be granted. Database, schema, and table are lower-level object types that do not support these privileges.Reference:Access Control Privileges | Snowflake Documentation,Using Dynamic Data Masking | Snowflake Documentation,Using Row Access Policies | Snowflake Documentation,Using Session Policies | Snowflake Documentation

Question 3

Question Type: MultipleChoice

A table for IOT devices that measures water usage is created. The table quickly becomes large and contains more than 2 billion rows.

```
create table water_iot (
UniqueId number,
DeviceId varchar(20),
DeviceManufacturer varchar(50)
CustomerId varchar(20),
IOT_timestamp timestamp_ntz,
City varchar(80),
Location varchar(50)
)
```

The general query patterns for the table are:

1. DeviceId, IOT_timestamp and CustomerId are frequently used in the filter predicate for the

select statement

2. The columns City and DeviceManuf acturer are often retrieved

3. There is often a count on UniqueId

Which field(s) should be used for the clustering key?

Options:

A- IOT_timestamp
B- City and DeviceManuf acturer
C- DeviceId and CustomerId
D- UniqueId

Answer:

C

Explanation:

A clustering key is a subset of columns or expressions that are used to co-locate the data in the same micro-partitions, which are the units of storage in Snowflake. Clustering can improve the performance of queries that filter on the clustering key columns, as it reduces the amount of data that needs to be scanned. The best choice for a clustering key depends on the query patterns and the data distribution in the table. In this case, the columns DeviceId, IOT_timestamp, and CustomerId are frequently used in the filter predicate for the select statement, which means they are good candidates for the clustering key. The columns City and DeviceManufacturer are often retrieved, but not filtered on, so they are not as important for the clustering key. The column UniqueId is used for counting, but it is not a good choice for the clustering key, as it is likely to have a high cardinality and a uniform distribution, which means it will not help to co-locate the data. Therefore, the best option is to use DeviceId and CustomerId as the clustering key, as they can help to prune the micro-partitions and speed up the queries.Reference:Clustering Keys & Clustered Tables,Micro-partitions & Data Clustering,A Complete Guide to Snowflake Clustering

Question 4

Question Type: MultipleChoice

What does a Snowflake Architect need to consider when implementing a Snowflake Connector for Kafka?

Options:

A- Every Kafka message is in JSON or Avro format.
B- The default retention time for Kafka topics is 14 days.
C- The Kafka connector supports key pair authentication, OAUTH. and basic authentication (for example, username and password).
D- The Kafka connector will create one table and one pipe to ingest data for each topic. If the connector cannot create the table or the pipe it will result in an exception.

Answer:

D

Explanation:

The Snowflake Connector for Kafka is a Kafka Connect sink connector that reads data from one or more Apache Kafka topics and loads the data into a Snowflake table. The connector supports different authentication methods to connect to Snowflake, such as key pair authentication, OAUTH, and basic authentication (for example, username and password).The connector also supports different encryption methods, such as HTTPS and SSL1.The connector does not require that every Kafka message is in JSON or Avro format, as it can handle other formats such as CSV, XML, and Parquet2. The default retention time for Kafka topics is not relevant for the connector, as it only consumes the messages that are available in the topics and does not store them in Kafka.The connector will create one table and one pipe to ingest data for each topic by default, but this behavior can be customized by using the snowflake.topic2table.map configuration property3.If the connector cannot create the table or the pipe, it will log an error and retry the operation until it succeeds or the connector is stopped4.Reference:

Installing and Configuring the Kafka Connector

Overview of the Kafka Connector

Managing the Kafka Connector

Troubleshooting the Kafka Connector

Question 5

Question Type: MultipleChoice

Company A would like to share data in Snowflake with Company B. Company B is not on the same cloud platform as Company A.

What is required to allow data sharing between these two companies?

Options:

A- Create a pipeline to write shared data to a cloud storage location in the target cloud provider.
B- Ensure that all views are persisted, as views cannot be shared across cloud platforms.
C- Setup data replication to the region and cloud platform where the consumer resides.
D- Company A and Company B must agree to use a single cloud platform: Data sharing is only possible if the companies share the same cloud provider.

Answer:

C

Explanation:

According to the SnowPro Advanced: Architect documents and learning resources, the requirement to allow data sharing between two companies that are not on the same cloud platform is to set up data replication to the region and cloud platform where the consumer resides. Data replication is a feature of Snowflake that enables copying databases across accounts in different regions and cloud platforms. Data replication allows data providers to securely share data with data consumers across different regions and cloud platforms by creating a replica database in the consumer's account. The replica database is read-only and automatically synchronized with the primary database in the provider's account.Data replication is useful for scenarios where data sharing is not possible or desirable due to latency, compliance, or security reasons1. The other options are incorrect because they are not required or feasible to allow data sharing between two companies that are not on the same cloud platform. Option A is incorrect because creating a pipeline to write shared data to a cloud storage location in the target cloud provider is not a secure or efficient way of sharing data. It would require additional steps to load the data from the cloud storage to the consumer's account, and it would not leverage the benefits of Snowflake's data sharing features. Option B is incorrect because ensuring that all views are persisted is not relevant for data sharing across cloud platforms. Views can be shared across cloud platforms as long as they reference objects in the same database.Persisting views is an option to improve the performance of querying views, but it is not required for data sharing2. Option D is incorrect because Company A and Company B do not need to agree to use a single cloud platform.Data sharing is possible across different cloud platforms using data replication or other methods, such as listings or auto-fulfillment3.Reference:Replicating Databases Across Multiple Accounts | Snowflake Documentation,Persisting Views | Snowflake Documentation,Sharing Data Across Regions and Cloud Platforms | Snowflake Documentation

Question 6

Question Type: MultipleChoice

An Architect on a new project has been asked to design an architecture that meets Snowflake security, compliance, and governance requirements as follows:

1) Use Tri-Secret Secure in Snowflake

2) Share some information stored in a view with another Snowflake customer

3) Hide portions of sensitive information from some columns

4) Use zero-copy cloning to refresh the non-production environment from the production environment

To meet these requirements, which design elements must be implemented? (Choose three.)

Options:

A- Define row access policies.
B- Use the Business-Critical edition of Snowflake.
C- Create a secure view.
D- Use the Enterprise edition of Snowflake.
E- Use Dynamic Data Masking.
F- Create a materialized view.

Answer:

B, C, E

Explanation:

These three design elements are required to meet the security, compliance, and governance requirements for the project.

To use Tri-Secret Secure in Snowflake, the Business Critical edition of Snowflake is required. This edition provides enhanced data protection features, such as customer-managed encryption keys, that are not available in lower editions.Tri-Secret Secure is a feature that combines a Snowflake-maintained key and a customer-managed key to create a composite master key to encrypt the data in Snowflake1.

To share some information stored in a view with another Snowflake customer, a secure view is recommended. A secure view is a view that hides the underlying data and the view definition from unauthorized users.Only the owner of the view and the users who are granted the owner's

role can see the view definition and the data in the base tables of the view2.A secure view can be shared with another Snowflake account using a data share3.

To hide portions of sensitive information from some columns, Dynamic Data Masking can be used. Dynamic Data Masking is a feature that allows applying masking policies to columns to selectively mask plain-text data at query time.Depending on the masking policy conditions and the user's role, the data can be fully or partially masked, or shown as plain-text4.

Question 7

Question Type: MultipleChoice

An Architect needs to design a Snowflake account and database strategy to store and analyze large amounts of structured and semi-structured dat

a. There are many business units and departments within the company. The requirements are scalability, security, and cost efficiency.

What design should be used?

Options:

A- Create a single Snowflake account and database for all data storage and analysis needs, regardless of data volume or complexity.
B- Set up separate Snowflake accounts and databases for each department or business unit, to ensure data isolation and security.
C- Use Snowflake's data lake functionality to store and analyze all data in a central location, without the need for structured schemas or indexes
D- Use a centralized Snowflake database for core business data, and use separate databases for departmental or project-specific data.

Answer:

D

Explanation:

The best design to store and analyze large amounts of structured and semi-structured data for different business units and departments is to use a centralized Snowflake database for core business data, and use separate databases for departmental or project-specific data. This design allows for scalability, security, and cost efficiency by leveraging Snowflake's features such as:

Database cloning:Cloning a database creates a zero-copy clone that shares the same data files as the original database, but can be modified independently. This reduces storage costs and enables fast and consistent data replication for different purposes.

Database sharing:Sharing a database allows granting secure and governed access to a subset of data in a database to other Snowflake accounts or consumers. This enables data collaboration and monetization across different business units or external partners.

Warehouse scaling:Scaling a warehouse allows adjusting the size and concurrency of a warehouse to match the performance and cost requirements of different workloads. This enables optimal resource utilization and flexibility for different data analysis needs.Reference:Snowflake Documentation: Database Cloning,Snowflake Documentation: Database Sharing, [Snowflake Documentation: Warehouse Scaling]

Question 8

Question Type: MultipleChoice

An Architect is integrating an application that needs to read and write data to Snowflake without installing any additional software on the application server.

How can this requirement be met?

Options:

A- Use SnowSQL.
B- Use the Snowpipe REST API.
C- Use the Snowflake SQL REST API.
D- Use the Snowflake ODBC driver.

Answer:

C

Explanation:

The Snowflake SQL REST API is a REST API that you can use to access and update data in a Snowflake database. You can use this API to execute standard queries and most DDL and DML statements. This API can be used to develop custom applications and integrations that can read and write data to Snowflake without installing any additional software on the application server. Option A is not correct because SnowSQL is a command-line client that requires installation and configuration on the application server. Option B is not correct because the Snowpipe REST API is

used to load data from cloud storage into Snowflake tables, not to read or write data to Snowflake. Option D is not correct because the Snowflake ODBC driver is a software component that enables applications to connect to Snowflake using the ODBC protocol, which also requires installation and configuration on the application server.Reference: The answer can be verified from Snowflake's official documentation on the Snowflake SQL REST API available on their website. Here are some relevant links:

Snowflake SQL REST API | Snowflake Documentation

Introduction to the SQL API | Snowflake Documentation

Submitting a Request to Execute SQL Statements | Snowflake Documentation

Question 9

Question Type: MultipleChoice

A company's daily Snowflake workload consists of a huge number of concurrent queries triggered between 9pm and 11pm. At the individual level, these queries are smaller statements that get completed within a short time period.

What configuration can the company's Architect implement to enhance the performance of this workload? (Choose two.)

Options:

A- Enable a multi-clustered virtual warehouse in maximized mode during the workload duration.
B- Set the MAX_CONCURRENCY_LEVEL to a higher value than its default value of 8 at the virtual warehouse level.
C- Increase the size of the virtual warehouse to size X-Large.
D- Reduce the amount of data that is being processed through this workload.
E- Set the connection timeout to a higher value than its default.

Answer:

A, B

Explanation:

These two configuration options can enhance the performance of the workload that consists of a huge number of concurrent queries that are smaller and faster.

Enabling a multi-clustered virtual warehouse in maximized mode allows the warehouse to scale out automatically by adding more clusters as soon as the current cluster is fully loaded, regardless of the number of queries in the queue. This can improve the concurrency and throughput of the workload by minimizing or preventing queuing.The maximized mode is suitable for workloads that require high performance and low latency, and are less sensitive to credit consumption1.

Setting the MAX_CONCURRENCY_LEVEL to a higher value than its default value of 8 at the virtual warehouse level allows the warehouse to run more queries concurrently on each cluster. This can improve the utilization and efficiency of the warehouse resources, especially for smaller and faster queries that do not require a lot of processing power.The MAX_CONCURRENCY_LEVEL parameter can be set when creating or modifying a warehouse, and it can be changed at any time2.

Snowflake Documentation: Scaling Policy for Multi-cluster Warehouses

Snowflake Documentation: MAX_CONCURRENCY_LEVEL

Question 10

Question Type: MultipleChoice

A company is using Snowflake in Azure in the Netherlands. The company analyst team also has data in JSON format that is stored in an Amazon S3 bucket in the AWS Singapore region that the team wants to analyze.

The Architect has been given the following requirements:

1. Provide access to frequently changing data

2. Keep egress costs to a minimum

3. Maintain low latency

How can these requirements be met with the LEAST amount of operational overhead?

Options:

A- Use a materialized view on top of an external table against the S3 bucket in AWS Singapore.
B- Use an external table against the S3 bucket in AWS Singapore and copy the data into transient tables.
C- Copy the data between providers from S3 to Azure Blob storage to collocate, then use Snowpipe for data ingestion.
D- Use AWS Transfer Family to replicate data between the S3 bucket in AWS Singapore and an

Azure Netherlands Blob storage, then use an external table against the Blob storage.

Answer:

B

Explanation:

:Option A is the best design to meet the requirements because it uses a materialized view on top of an external table against the S3 bucket in AWS Singapore.A materialized view is a database object that contains the results of a query and can be refreshed periodically to reflect changes in the underlying data1.An external table is a table that references data files stored in a cloud storage service, such as Amazon S32. By using a materialized view on top of an external table, the company can provide access to frequently changing data, keep egress costs to a minimum, and maintain low latency. This is because the materialized view will cache the query results in Snowflake, reducing the need to access the external data files and incur network charges. The materialized view will also improve the query performance by avoiding scanning the external data files every time.The materialized view can be refreshed on a schedule or on demand to capture the changes in the external data files1.

Option B is not the best design because it uses an external table against the S3 bucket in AWS Singapore and copies the data into transient tables.A transient table is a table that is not subject to the Time Travel and Fail-safe features of Snowflake, and is automatically purged after a period of time3. By using an external table and copying the data into transient tables, the company will incur more egress costs and operational overhead than using a materialized view. This is because the external table will access the external data files every time a query is executed, and the copy operation will also transfer data from S3 to Snowflake. The transient tables will also consume more storage space in Snowflake and require manual maintenance to ensure they are up to date.

Option C is not the best design because it copies the data between providers from S3 to Azure Blob storage to collocate, then uses Snowpipe for data ingestion.Snowpipe is a service that automates the loading of data from external sources into Snowflake tables4. By copying the data between providers, the company will incur high egress costs and latency, as well as operational complexity and maintenance of the infrastructure. Snowpipe will also add another layer of processing and storage in Snowflake, which may not be necessary if the external data files are already in a queryable format.

Option D is not the best design because it uses AWS Transfer Family to replicate data between the S3 bucket in AWS Singapore and an Azure Netherlands Blob storage, then uses an external table against the Blob storage.AWS Transfer Family is a service that enables secure and seamless transfer of files over SFTP, FTPS, and FTP to and from Amazon S3 or Amazon EFS5. By using AWS Transfer Family, the company will incur high egress costs and latency, as well as operational complexity and maintenance of the infrastructure. The external table will also access the external data files every time a query is executed, which may affect the query performance.