Snowflake ARA-C01 Practice Questions

Shared by Barker on 19-08-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

An Architect is troubleshooting a query with poor performance using the QUERY function. The Architect observes that the COMPILATION_TIME Is greater than the EXECUTION_TIME.

What is the reason for this?

Options:

A- The query is processing a very large dataset.
B- The query has overly complex logic.
C- The query Is queued for execution.
D- The query Is reading from remote storage

Answer:

B

Explanation:

The correct answer is B because the compilation time is the time it takes for the optimizer to create an optimal query plan for the efficient execution of the query. The compilation time depends on the complexity of the query, such as the number of tables, columns, joins, filters, aggregations, subqueries, etc. The more complex the query, the longer it takes to compile.

Option A is incorrect because the query processing time is not affected by the size of the dataset, but by the size of the virtual warehouse. Snowflake automatically scales the compute resources to match the data volume and parallelizes the query execution. The size of the dataset may affect the execution time, but not the compilation time.

Option C is incorrect because the query queue time is not part of the compilation time or the execution time. It is a separate metric that indicates how long the query waits for a warehouse slot before it starts running. The query queue time depends on the warehouse load, concurrency, and priority settings.

Option D is incorrect because the query remote IO time is not part of the compilation time or the execution time. It is a separate metric that indicates how long the query spends reading data from remote storage, such as S3 or Azure Blob Storage. The query remote IO time depends on the network latency, bandwidth, and caching efficiency.Reference:

Understanding Why Compilation Time in Snowflake Can Be Higher than Execution Time: This article explains why the total duration (compilation + execution) time is an essential metric to measure query performance in Snowflake. It discusses the reasons for the long compilation time, including query complexity and the number of tables and columns.

Exploring Execution Times: This document explains how to examine the past performance of queries and tasks using Snowsight or by writing queries against views in the ACCOUNT_USAGE schema. It also describes the different metrics and dimensions that affect query performance, such as duration, compilation, execution, queue, and remote IO time.

What is the ''compilation time'' and how to optimize it?: This community post provides some tips and best practices on how to reduce the compilation time, such as simplifying the query logic, using views or common table expressions, and avoiding unnecessary columns or joins.

Question 2

Question Type: MultipleChoice

Database DB1 has schema S1 which has one table, T1.

DB1 --> S1 --> T1

The retention period of EG1 is set to 10 days.

The retention period of s: is set to 20 days.

The retention period of t: Is set to 30 days.

The user runs the following command:

Drop Database DB1;

What will the Time Travel retention period be for T1?

Options:

A- 10 days
B- 20 days
C- 30 days
D- 37 days

Answer:

C

Explanation:

The Time Travel retention period for T1 will be 30 days, which is the retention period set at the table level. The Time Travel retention period determines how long the historical data is

preserved and accessible for an object after it is modified or dropped. The Time Travel retention period can be set at the account level, the database level, the schema level, or the table level. The retention period set at the lowest level of the hierarchy takes precedence over the higher levels. Therefore, the retention period set at the table level overrides the retention periods set at the schema level, the database level, or the account level. When the user drops the database DB1, the table T1 is also dropped, but the historical data is still preserved for 30 days, which is the retention period set at the table level. The user can use the UNDROP command to restore the table T1 within the 30-day period. The other options are incorrect because:

10 days is the retention period set at the database level, which is overridden by the table level.

20 days is the retention period set at the schema level, which is also overridden by the table level.

37 days is not a valid option, as it is not the retention period set at any level.

Understanding & Using Time Travel

AT | BEFORE

Snowflake Time Travel & Fail-safe

Question 3

Question Type: MultipleChoice

An Architect is troubleshooting a query with poor performance using the QUERY function. The Architect observes that the COMPILATION_TIME Is greater than the EXECUTION_TIME.

What is the reason for this?

Options:

A- The query is processing a very large dataset.
B- The query has overly complex logic.
C- The query Is queued for execution.
D- The query Is reading from remote storage

Answer:

B

Explanation:

The correct answer is B because the compilation time is the time it takes for the optimizer to create an optimal query plan for the efficient execution of the query. The compilation time depends on the complexity of the query, such as the number of tables, columns, joins, filters, aggregations, subqueries, etc. The more complex the query, the longer it takes to compile.

Option A is incorrect because the query processing time is not affected by the size of the dataset, but by the size of the virtual warehouse. Snowflake automatically scales the compute resources to match the data volume and parallelizes the query execution. The size of the dataset may affect the execution time, but not the compilation time.

Option C is incorrect because the query queue time is not part of the compilation time or the execution time. It is a separate metric that indicates how long the query waits for a warehouse slot before it starts running. The query queue time depends on the warehouse load, concurrency, and priority settings.

Option D is incorrect because the query remote IO time is not part of the compilation time or the execution time. It is a separate metric that indicates how long the query spends reading data from remote storage, such as S3 or Azure Blob Storage. The query remote IO time depends on the network latency, bandwidth, and caching efficiency.Reference:

Understanding Why Compilation Time in Snowflake Can Be Higher than Execution Time: This article explains why the total duration (compilation + execution) time is an essential metric to measure query performance in Snowflake. It discusses the reasons for the long compilation time, including query complexity and the number of tables and columns.

Exploring Execution Times: This document explains how to examine the past performance of queries and tasks using Snowsight or by writing queries against views in the ACCOUNT_USAGE schema. It also describes the different metrics and dimensions that affect query performance, such as duration, compilation, execution, queue, and remote IO time.

What is the ''compilation time'' and how to optimize it?: This community post provides some tips and best practices on how to reduce the compilation time, such as simplifying the query logic, using views or common table expressions, and avoiding unnecessary columns or joins.

Question 4

Question Type: MultipleChoice

What integration object should be used to place restrictions on where data may be exported?

Options:

A- Stage integration
B- Security integration
C- Storage integration
D- API integration

Answer:

C

Explanation:

In Snowflake, a storage integration is used to define and configure external cloud storage that Snowflake will interact with. This includes specifying security policies for access control. One of the main features of storage integrations is the ability to set restrictions on where data may be exported. This is done by binding the storage integration to specific cloud storage locations, thereby ensuring that Snowflake can only access those locations. It helps to maintain control over the data and complies with data governance and security policies by preventing unauthorized data exports to unspecified locations.

Question 5

Question Type: MultipleChoice

A new user user_01 is created within Snowflake. The following two commands are executed:

Command 1-> show grants to user user_01;

Command 2 ~> show grants on user user 01;

What inferences can be made about these commands?

Options:

A- Command 1 defines which user owns user_01
Command 2 defines all the grants which have been given to user_01
B- Command 1 defines all the grants which are given to user_01 Command 2 defines which user owns user_01
C- Command 1 defines which role owns user_01
Command 2 defines all the grants which have been given to user_01
D- Command 1 defines all the grants which are given to user_01
Command 2 defines which role owns user 01

Answer:

D

Explanation:

The SHOW GRANTS command in Snowflake can be used to list all the access control privileges that have been explicitly granted to roles, users, and shares.The syntax and the output of the command vary depending on the object type and the grantee type specified in the command1. In this question, the two commands have the following meanings:

Command 1: show grants to user user_01; This command lists all the roles granted to the user user_01. The output includes the role name, the grantee name, and the granted by role name for each grant.This command is equivalent to show grants to user current_user if user_01 is the current user1.

Command 2: show grants on user user_01; This command lists all the privileges that have been granted on the user object user_01. The output includes the privilege name, the grantee name, and the granted by role name for each grant.This command shows which role owns the user object user_01, as the owner role has the privilege to modify or drop the user object2.

Therefore, the correct inference is that command 1 defines all the grants which are given to user_01, and command 2 defines which role owns user_01.

SHOW GRANTS

Understanding Access Control in Snowflake

Question 6

Question Type: MultipleChoice

How do Snowflake databases that are created from shares differ from standard databases that are not created from shares? (Choose three.)

Options:

A- Shared databases are read-only.
B- Shared databases must be refreshed in order for new data to be visible.
C- Shared databases cannot be cloned.
D- Shared databases are not supported by Time Travel.
E- Shared databases will have the PUBLIC or INFORMATION_SCHEMA schemas without explicitly granting these schemas to the share.
F- Shared databases can also be created as transient databases.

Answer:

A, C, D

Explanation:

According to the SnowPro Advanced: Architect documents and learning resources, the ways that Snowflake databases that are created from shares differ from standard databases that are not created from shares are:

Shared databases are read-only. This means that the data consumers who access the shared databases cannot modify or delete the data or the objects in the databases.The data providers who share the databases have full control over the data and the objects, and can grant or revoke privileges on them1.

Shared databases cannot be cloned. This means that the data consumers who access the shared databases cannot create a copy of the databases or the objects in the databases.The data providers who share the databases can clone the databases or the objects, but the clones are not automatically shared2.

Shared databases are not supported by Time Travel. This means that the data consumers who access the shared databases cannot use the AS OF clause to query historical data or restore deleted data.The data providers who share the databases can use Time Travel on the databases or the objects, but the historical data is not visible to the data consumers3.

The other options are incorrect because they are not ways that Snowflake databases that are created from shares differ from standard databases that are not created from shares. Option B is incorrect because shared databases do not need to be refreshed in order for new data to be visible.The data consumers who access the shared databases can see the latest data as soon as the data providers update the data1. Option E is incorrect because shared databases will not have the PUBLIC or INFORMATION_SCHEMA schemas without explicitly granting these schemas to the share.The data consumers who access the shared databases can only see the objects that the data providers grant to the share, and the PUBLIC and INFORMATION_SCHEMA schemas are not granted by default4. Option F is incorrect because shared databases cannot be created as transient databases. Transient databases are databases that do not support Time Travel or Fail-safe, and can be dropped without affecting the retention period of the data.Shared databases are always created as permanent databases, regardless of the type of the source database5.Reference:Introduction to Secure Data Sharing | Snowflake Documentation,Cloning Objects | Snowflake Documentation,Time Travel | Snowflake Documentation,Working with Shares | Snowflake Documentation,CREATE DATABASE | Snowflake Documentation

Question 7

Question Type: MultipleChoice

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

Options:

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.

Answer:

A, B, C

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.

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