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.geekcert.com/ara-c01.html

100% Passing Guarantee
100% Money Back Assurance

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

- **Instant Download** After Purchase
- **100% Money Back** Guarantee
- **365 Days** Free Update
- **800,000+** Satisfied Customers

QUESTION 1

A table contains five columns and it has millions of records. The cardinality distribution of the columns is shown below:

Column	Number of Distinct Values
C1	10,790
C2	108
C3	302,605
C4	1.117,736
C5	2.205,400

Column C4 and C5 are mostly used by SELECT queries in the GROUP BY and ORDER BY clauses. Whereas columns C1, C2 and C3 are heavily used in filter and join conditions of SELECT queries.

The Architect must design a clustering key for this table to improve the query performance.

Based on Snowflake recommendations, how should the clustering key columns be ordered while defining the multi-column clustering key?

A. C5, C4, C2

B. C3, C4, C5

C. C1, C3, C2

D. C2, C1, C3

Correct Answer: C

Explanation: According to the Snowflake documentation, the following are some considerations for choosing clustering for a table1:

Clustering is optimal when either:

Clustering is most effective when the clustering key is used in the following types of query predicates:

Clustering is less effective when the clustering key is not used in any of the above query predicates, or when the clustering key is used in a predicate that requires a function or expression to be applied to the key (e.g. DATE_TRUNC,

TO_CHAR, etc.).

For most tables, Snowflake recommends a maximum of 3 or 4 columns (or expressions) per key. Adding more than 3-4 columns tends to increase costs more than benefits.

Based on these considerations, the best option for the clustering key columns is C. C1, C3, C2, because:

These columns are heavily used in filter and join conditions of SELECT queries, which are the most effective types of

predicates for clustering. These columns have high cardinality, which means they have many distinct values and can help

reduce the clustering skew and improve the compression ratio. These columns are likely to be correlated with each other, which means they can help co-locate similar rows in the same micro-partitions and improve the scan efficiency.

These columns do not require any functions or expressions to be applied to them, which means they can be directly used in the predicates without affecting the clustering.

References: 1: Considerations for Choosing Clustering for a Table | Snowflake Documentation

QUESTION 2

Which of the following are characteristics of Snowflake\\'s parameter hierarchy?

A. Session parameters override virtual warehouse parameters.

B. Virtual warehouse parameters override user parameters.

C. Table parameters override virtual warehouse parameters.

D. Schema parameters override account parameters.

Correct Answer: D

Explanation: This is the correct answer because it reflects the characteristics of Snowflake\\'s parameter hierarchy. Snowflake provides three types of parameters that can be set for an account: account parameters, session parameters, and object parameters. All parameters have default values, which can be set and then overridden at different levels depending on the parameter type. The following diagram illustrates the hierarchical relationship between the different parameter types and how individual parameters can be overridden at each level1: As shown in the diagram, schema parameters are a type of object parameters that can be set for schemas. Schema parameters can override the account parameters that are set at the account level. For example, the LOG_LEVEL parameter can be set at the account level to control the logging level for all objects in the account, but it can also be overridden at the schema level to control the logging level for specific stored procedures and UDFs in that schema2. The other options listed are not correct because they do not reflect the characteristics of Snowflake\\'s parameter hierarchy. Session parameters do not override virtual warehouse parameters, because virtual warehouse parameters are a type of session parameters that can be set for virtual warehouses. Virtual warehouse parameters do not override user parameters, because user parameters are a type of session parameters that can be set for users. Table parameters do not override virtual warehouse parameters, because table parameters are a type of object parameters that can be set for tables, and object parameters do not affect session parameters1. References: Snowflake Documentation: Parameters Snowflake Documentation: Setting Log Level

QUESTION 3

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 4

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 5

The Data Engineering team at a large manufacturing company needs to engineer data coming from many sources to support a wide variety of use cases and data consumer requirements which include:

1) Finance and Vendor Management team members who require reporting and visualization

2) Data Science team members who require access to raw data for ML model development

3) Sales team members who require engineered and protected data for data monetization

What Snowflake data modeling approaches will meet these requirements? (Choose two.)

A. Consolidate data in the company\\'s data lake and use EXTERNAL TABLES.

B. Create a raw database for landing and persisting raw data entering the data pipelines.

C. Create a set of profile-specific databases that aligns data with usage patterns.

D. Create a single star schema in a single database to support all consumers\\' requirements.

E. Create a Data Vault as the sole data pipeline endpoint and have all consumers directly access the Vault.

Correct Answer: BC

Explanation: These two approaches are recommended by Snowflake for data modeling in a data lake scenario. Creating a raw database allows the data engineering team to ingest data from various sources without any transformation or cleansing, preserving the original data quality and format. This enables the data science team to access the raw data for ML model development. Creating a set of profile-specific databases allows the data engineering team to apply different transformations and optimizations for different use cases and data consumer requirements. For example, the finance and vendor management team can access a dimensional database that supports reporting and visualization, while the sales team can access a secure database that supports data monetization. References: Snowflake Data Lake Architecture | Snowflake Documentation Snowflake Data Lake Best Practices | Snowflake Documentation