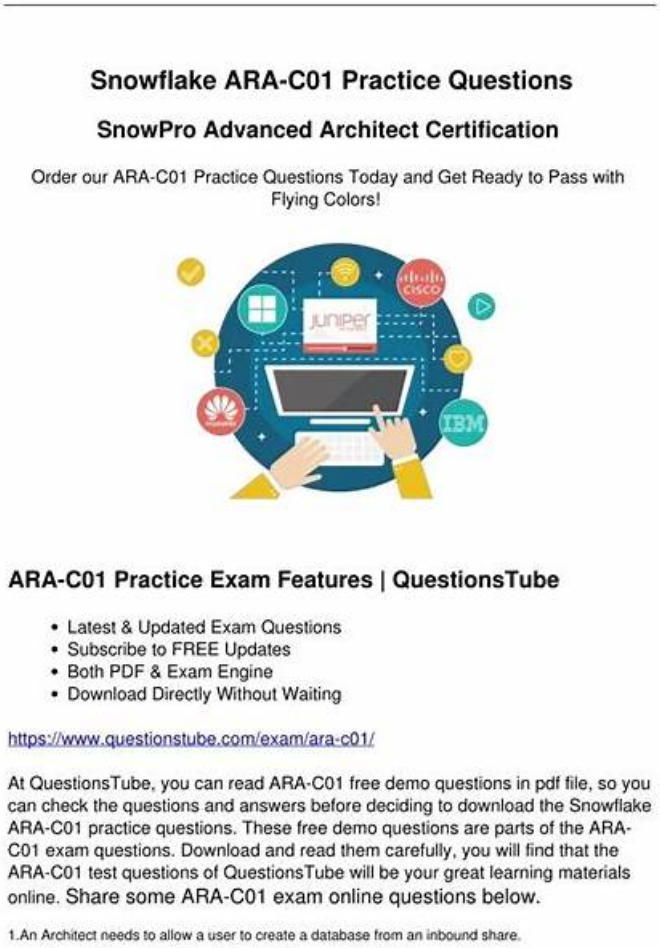
Snowflake ARA-C01 Training Solutions & New Braindumps ARA-C01 Book



BTW, DOWNLOAD part of PassCollection ARA-C01 dumps from Cloud Storage: https://drive.google.com/open?id=1FH4O_DD_qf2vCKohLGp1-P8JHr7lo_1y

ARA-C01 exam questions have a very high hit rate, of course, will have a very high pass rate. Before you select a product, you must have made a comparison of your own pass rates. Our ARA-C01 study materials must appear at the top of your list. And our ARA-C01 learning quiz has a 99% pass rate. This is the result of our efforts and the best gift to the user. And it is also proved and tested the quality of our ARA-C01 training engine is excellent.

Snowflake ARA-C01 Certification Exam covers a wide range of topics including Snowflake architecture, data modeling, performance optimization, security, and governance. To pass the certification exam, candidates must demonstrate their expertise in designing and implementing Snowflake solutions that meet the business requirements of their organization. Candidates must also demonstrate their ability to optimize the performance of Snowflake solutions and ensure the security and governance of Snowflake data.

>> Snowflake ARA-C01 Training Solutions <<

New Braindumps ARA-C01 Book - Reliable ARA-C01 Test Pass4sure

We have always taken care to provide the best Snowflake ARA-C01 exam dumps to our customers. That's why we offer many

other benefits with our product. We provide a demo version of the real product to our customers to clear their doubts about the truthfulness and accuracy of SnowPro Advanced Architect Certification (ARA-C01) preparation material. You can try the product before you buy it.

Snowflake SnowPro Advanced Architect Certification Sample Questions (Q123-Q128):

NEW QUESTION # 123
An Architect runs the following SQL query:
How can this query be interpreted?

- A. FILERONS is the file format location. FILE_ROW_NUMBER is a stage.
- B. FILEROWS is a stage. FILE_ROW_NUMBER is line number in file.
- C. FILEROWS is the table. FILE_ROW_NUMBER is the line number in the table.
- D. FILEROWS is a file. FILE_ROW_NUMBER is the file format location.

Answer: B

Explanation:
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.
Stages
LIST Function

NEW QUESTION # 124
Who can provide permission to EXECUTE TASK?

- A. THE TASK OWNER
- B. ACCOUNTADMIN
- C. SYSADMIN

Answer: B

NEW QUESTION # 125
Assuming all Snowflake accounts are using an Enterprise edition or higher, in which development and testing scenarios would be copying of data be required, and zero-copy cloning not be suitable? (Select TWO).

- A. Data is in a production Snowflake account that needs to be provided to Developers in a separate development/testing Snowflake account in the same cloud region.
- B. The release process requires pre-production testing of changes with data of production scale and complexity. For security reasons, pre-production also runs in the production account.
- C. Developers create their own datasets to work against transformed versions of the live data.
- D. Production and development run in different databases in the same account, and Developers need to see production-like data but with specific columns masked.
- E. Developers create their own copies of a standard test database previously created for them in the development account, for their initial development and unit testing.

Answer: A,C

Explanation:
Zero-copy cloning is a feature that allows creating a clone of a table, schema, or database without physically copying the data. Zero-copy cloning is suitable for scenarios where the cloned object needs to have the same data and metadata as the original object, and where the cloned object does not need to be modified or updated frequently. Zero-copy cloning is also suitable for scenarios where the cloned object needs to be shared within the same Snowflake account or across different accounts in the same cloud region2

However, zero-copy cloning is not suitable for scenarios where the cloned object needs to have different data or metadata than the original object, or where the cloned object needs to be modified or updated frequently. Zero-copy cloning is also not suitable for scenarios where the cloned object needs to be shared across different accounts in different cloud regions. In these scenarios, copying of data would be required, either by using the COPY INTO command or by using data sharing with secure views3 The following are examples of development and testing scenarios where copying of data would be required, and zero-copy cloning would not be suitable:
Developers create their own datasets to work against transformed versions of the live data. This scenario requires copying of data because the developers need to modify the data or metadata of the cloned object to perform transformations, such as adding, deleting, or updating columns, rows, or values. Zero-copy cloning would not be suitable because it would create a read-only clone that shares the same data and metadata as the original object, and any changes made to the clone would affect the original object as well4 Data is in a production Snowflake account that needs to be provided to Developers in a separate development/testing Snowflake account in the same cloud region. This scenario requires copying of data because the data needs to be shared across different accounts in the same cloud region. Zero-copy cloning would not be suitable because it would create a clone within the same account as the original object, and it would not allow sharing the clone with another account. To share data across different accounts in the same cloud region, data sharing with secure views or COPY INTO command can be used5 The following are examples of development and testing scenarios where zero-copy cloning would be suitable, and copying of data would not be required:
Production and development run in different databases in the same account, and Developers need to see production-like data but with specific columns masked. This scenario can use zero-copy cloning because the data needs to be shared within the same account, and the cloned object does not need to have different data or metadata than the original object. Zero-copy cloning can create a clone of the production database in the development database, and the clone can have the same data and metadata as the original database. To mask specific columns, secure views can be created on top of the clone, and the developers can access the secure views instead of the clone directly6 Developers create their own copies of a standard test database previously created for them in the development account, for their initial development and unit testing. This scenario can use zero-copy cloning because the data needs to be shared within the same account, and the cloned object does not need to have different data or metadata than the original object. Zero-copy cloning can create a clone of the standard test database for each developer, and the clone can have the same data and metadata as the original database. The developers can use the clone for their initial development and unit testing, and any changes made to the clone would not affect the original database or other clones7 The release process requires pre-production testing of changes with data of production scale and complexity. For security reasons, pre-production also runs in the production account. This scenario can use zero-copy cloning because the data needs to be shared within the same account, and the cloned object does not need to have different data or metadata than the original object. Zero-copy cloning can create a clone of the production database in the pre-production database, and the clone can have the same data and metadata as the original database. The pre-production testing can use the clone to test the changes with data of production scale and complexity, and any changes made to the clone would not affect the original database or the production environment8 Reference:
1: SnowPro Advanced: Architect | Study Guide 9
2: Snowflake Documentation | Cloning Overview
3: Snowflake Documentation | Loading Data Using COPY into a Table
4: Snowflake Documentation | Transforming Data During a Load
5: Snowflake Documentation | Data Sharing Overview
6: Snowflake Documentation | Secure Views
7: Snowflake Documentation | Cloning Databases, Schemas, and Tables
8: Snowflake Documentation | Cloning for Testing and Development
9: SnowPro Advanced: Architect | Study Guide
10: Cloning Overview
11: Loading Data Using COPY into a Table
12: Transforming Data During a Load
13: Data Sharing Overview
14: Secure Views
15: Cloning Databases, Schemas, and Tables
16: Cloning for Testing and Development

NEW QUESTION # 126
A company has a source system that provides JSON records for various IoT operations. The JSON Is loading directly into a persistent table with a variant field. The data Is quickly growing to 100s of millions of records and performance to becoming an issue. There is a generic access pattern that Is used to filter on the create_date key within the variant field.
What can be done to improve performance?

- A. Incorporate the use of multiple tables partitioned by date ranges. When a user or process needs to query a particular date range, ensure the appropriate base table Is used.
- B. Alter the target table to Include additional fields pulled from the JSON records. This would Include a create_date field with

a datatype of time stamp. When this field Is used in the filter, partition pruning will occur.

- C. Validate the size of the warehouse being used. If the record count is approaching 100s of millions, size XL will be the minimum size required to process this amount of data.
- D. Alter the target table to include additional fields pulled from the JSON records. This would include a create_date field with a datatype of varchar. When this field is used in the filter, partition pruning will occur.

Answer: B

Explanation:
The correct answer is A because it improves the performance of queries by reducing the amount of data scanned and processed. By adding a create_date field with a timestamp data type, Snowflake can automatically cluster the table based on this field and prune the micro-partitions that do not match the filter condition. This avoids the need to parse the JSON data and access the variant field for every record.
Option B is incorrect because it does not improve the performance of queries. By adding a create_date field with a varchar data type, Snowflake cannot automatically cluster the table based on this field and prune the micro-partitions that do not match the filter condition. This still requires parsing the JSON data and accessing the variant field for every record.
Option C is incorrect because it does not address the root cause of the performance issue. By validating the size of the warehouse being used, Snowflake can adjust the compute resources to match the data volume and parallelize the query execution. However, this does not reduce the amount of data scanned and processed, which is the main bottleneck for queries on JSON data.
Option D is incorrect because it adds unnecessary complexity and overhead to the data loading and querying process. By incorporating the use of multiple tables partitioned by date ranges, Snowflake can reduce the amount of data scanned and processed for queries that specify a date range. However, this requires creating and maintaining multiple tables, loading data into the appropriate table based on the date, and joining the tables for queries that span multiple date ranges. Reference:
Snowflake Documentation: Loading Data Using Snowpipe: This document explains how to use Snowpipe to continuously load data from external sources into Snowflake tables. It also describes the syntax and usage of the COPY INTO command, which supports various options and parameters to control the loading behavior, such as ON_ERROR, PURGE, and SKIP_FILE.
Snowflake Documentation: Date and Time Data Types and Functions: This document explains the different data types and functions for working with date and time values in Snowflake. It also describes how to set and change the session timezone and the system timezone.
Snowflake Documentation: Querying Metadata: This document explains how to query the metadata of the objects and operations in Snowflake using various functions, views, and tables. It also describes how to access the copy history information using the COPY_HISTORY function or the COPY_HISTORY view.
Snowflake Documentation: Loading JSON Data: This document explains how to load JSON data into Snowflake tables using various methods, such as the COPY INTO command, the INSERT command, or the PUT command. It also describes how to access and query JSON data using the dot notation, the FLATTEN function, or the LATERAL join.
Snowflake Documentation: Optimizing Storage for Performance: This document explains how to optimize the storage of data in Snowflake tables to improve the performance of queries. It also describes the concepts and benefits of automatic clustering, search optimization service, and materialized views.

NEW QUESTION # 127
How do Snowflake databases that are created from shares differ from standard databases that are not created from shares? (Choose three.)

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

Answer: A,D,E

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. References: Introduction to Secure Data Sharing | Snowflake Documentation, Cloning Objects | Snowflake Documentation, Time Travel | Snowflake Documentation, Working with Shares | Snowflake Documentation, CREATE DATABASE | Snowflake Documentation

NEW QUESTION # 128
.....

The price for ARA-C01 training materials are reasonable, and no matter you are an employee in the company or a student at school, you can afford it. Besides ARA-C01 exam materials are high quality and accuracy, therefore, you can pass the exam just one time. In order to strengthen your confidence for ARA-C01 Exam Braindumps, we are pass guarantee and money back guarantee. We will give you full refund if you fail to pass the exam. We offer you free update for one year for ARA-C01 training materials, and the update version will be sent to your email address automatically.

What's more, part of that PassCollection ARA-C01 dumps now are free: https://drive.google.com/open?id=1FH4O_DD_qf2vCKohLGp1-P8JHr7lo_1y