Snowflake ARA-C01 Exam | ARA-C01 Valid Test Guide - Reliable Planform of ARA-C01 Reliable Test Simulator

2026 Latest ActualTorrent ARA-C01 PDF Dumps and ARA-C01 Exam Engine Free Share: https://drive.google.com/open?id=19xVAHT86HpzdkkhyuRZxtDtYyY3EHA47

These SnowPro Advanced Architect Certification (ARA-C01) practice test covers all the topics of the ARA-C01 test and includes real ARA-C01 questions. If you are attempting the ARA-C01 examination for the first time, you will get an exact idea about the ARA-C01 exam and how you can clear it with flying colors. These Snowflake ARA-C01 Questions are available in desktop ARA-C01 practice exam software, web-based ARA-C01 practice test, and SnowPro Advanced Architect Certification (ARA-C01) dumps pdf format.

The Snowflake ARA-C01 Exam is intended for experienced Snowflake architects who have a deep understanding of Snowflake's architecture, data warehousing concepts, and best practices. It is also designed to test an individual's ability to design, implement, and manage complex Snowflake solutions that meet a variety of business needs.

>> ARA-C01 Valid Test Guide <<

Free PDF Quiz 2026 Snowflake Valid ARA-C01: SnowPro Advanced Architect Certification Valid Test Guide

Our ARA-C01 practice materials are your best choice for their efficiency in different aspects: first of all, do not need to wait, you can get them immediately if you pay for it and download as your wish. Clear-arranged content is our second advantage. Some exam candidates are prone to get anxious about the ARA-C01 Exam Questions, but with clear and points of necessary questions within our ARA-C01 study guide, you can master them effectively in limited time.

Snowflake ARA-C01 exam is a rigorous assessment of an individual's technical abilities and understanding of Snowflake's architecture. ARA-C01 exam consists of multiple-choice questions and covers a broad range of topics, including data modeling, security, performance tuning, and integration with other systems. ARA-C01 Exam also tests an individual's ability to design and implement advanced Snowflake solutions that meet specific business requirements.

Snowflake SnowPro Advanced Architect Certification Sample Questions (Q76-Q81):

NEW QUESTION # 76
A media company needs a data pipeline that will ingest customer review data into a Snowflake table, and apply some transformations. The company also needs to use Amazon Comprehend to do sentiment analysis and make the de-identified final data set available publicly for advertising companies who use different cloud providers in different regions.
The data pipeline needs to run continuously and efficiently as new records arrive in the object storage leveraging event notifications. Also, the operational complexity, maintenance of the infrastructure, including platform upgrades and security, and the development effort should be minimal.
Which design will meet these requirements?

- A. Ingest the data using copy into and use streams and tasks to orchestrate transformations. Export the data into Amazon S3 to do model inference with Amazon Comprehend and ingest the data back into a Snowflake table. Then create a listing in the Snowflake Marketplace to make the data available to other companies.
- B. Ingest the data using Snowpipe and use streams and tasks to orchestrate transformations. Create an external function to do model inference with Amazon Comprehend and write the final records to a Snowflake table. Then create a listing in the Snowflake Marketplace to make the data available to other companies.
- C. Ingest the data into Snowflake using Amazon EMR and PySpark using the Snowflake Spark connector. Apply transformations using another Spark job. Develop a python program to do model inference by leveraging the Amazon Comprehend text analysis API. Then write the results to a Snowflake table and create a listing in the Snowflake Marketplace to make the data available to other companies.
- D. Ingest the data using Snowpipe and use streams and tasks to orchestrate transformations. Export the data into Amazon S3 to do model inference with Amazon Comprehend and ingest the data back into a Snowflake table. Then create a listing in the Snowflake Marketplace to make the data available to other companies.

Answer: B

Explanation:
Option B is the best design to meet the requirements because it uses Snowpipe to ingest the data continuously and efficiently as new records arrive in the object storage, leveraging event notifications. Snowpipe is a service that automates the loading of data from external sources into Snowflake tables1. It also uses streams and tasks to orchestrate transformations on the ingested data. Streams are objects that store the change history of a table, and tasks are objects that execute SQL statements on a schedule or when triggered by another task2. Option B also uses an external function to do model inference with Amazon Comprehend and write the final records to a Snowflake table. An external function is a user-defined function that calls an external API, such as Amazon Comprehend, to perform computations that are not natively supported by Snowflake3.
Finally, option B uses the Snowflake Marketplace to make the de-identified final data set available publicly for advertising companies who use different cloud providers in different regions. The Snowflake Marketplace is a platform that enables data providers to list and share their data sets with data consumers, regardless of the cloud platform or region they use4.
Option A is not the best design because it uses copy into to ingest the data, which is not as efficient and continuous as Snowpipe. Copy into is a SQL command that loads data from files into a table in a single transaction. It also exports the data into Amazon S3 to do model inference with Amazon Comprehend, which adds an extra step and increases the operational complexity and maintenance of the infrastructure.
Option C is not the best design because it uses Amazon EMR and PySpark to ingest and transform the data, which also increases the operational complexity and maintenance of the infrastructure. Amazon EMR is a cloud service that provides a managed Hadoop framework to process and analyze large-scale data sets.
PySpark is a Python API for Spark, a distributed computing framework that can run on Hadoop. Option C also develops a python program to do model inference by leveraging the Amazon Comprehend text analysis API, which increases the development effort.
Option D is not the best design because it is identical to option A, except for the ingestion method. It still exports the data into Amazon S3 to do model inference with Amazon Comprehend, which adds an extra step and increases the operational complexity and maintenance of the infrastructure.
References: 1: Snowpipe Overview 2: Using Streams and Tasks to Automate Data Pipelines 3: External Functions Overview 4: Snowflake Data Marketplace Overview : [Loading Data Using COPY INTO] : [What is Amazon EMR?] : [PySpark Overview]

NEW QUESTION # 77
Which statement is not true about shared database?

- A. Shared databases can be re-shared with other accounts
- B. Time travel is not supported on a shared database
- C. Shared databases cannot be cloned
- D. Shared databases are read only

Answer: A

NEW QUESTION # 78
A company has a table with that has corrupted data, named Data. The company wants to recover the data as it was 5 minutes ago using cloning and Time Travel.
What command will accomplish this?

- A. CREATE TABLE Recover Data CLONE Data AT(TIME => -60*5);
- B. CREATE TABLE Recover_Data CLONE Data AT(OFFSET => -60*5);

- C. CREATE CLONE TABLE Recover_Data FROM Data AT(OFFSET => -60*5);
- D. CREATE CLONE Recover_Data FROM Data AT(OFFSET => -60*5);

Answer: B

Explanation:
This is the correct command to create a clone of the table Data as it was 5 minutes ago using cloning and Time Travel. Cloning is a feature that allows creating a copy of a database, schema, table, or view without duplicating the data or metadata. Time Travel is a feature that enables accessing historical data (i.e. data that has been changed or deleted) at any point within a defined period. To create a clone of a table at a point in time in the past, the syntax is:
CREATE TABLE <clone_name> CLONE <source_table> AT (OFFSET => <offset_in_seconds>); The OFFSET parameter specifies the time difference in seconds from the present time. A negative value indicates a point in the past. For example, -60*5 means 5 minutes ago. Alternatively, the TIMESTAMP parameter can be used to specify an exact timestamp in the past. The clone will contain the data as it existed in the source table at the specified point in time12.
References:
* Snowflake Documentation: Cloning Objects
* Snowflake Documentation: Cloning Objects at a Point in Time in the Past

NEW QUESTION # 79
Company A has recently acquired company B. The Snowflake deployment for company B is located in the Azure West Europe region.
As part of the integration process, an Architect has been asked to consolidate company B's sales data into company A's Snowflake account which is located in the AWS us-east-1 region.
How can this requirement be met?

- A. Build a custom data pipeline using Azure Data Factory or a similar tool to extract the sales data from company B's Snowflake account. Transform the data, then load it into company A's Snowflake account.
- B. Replicate the sales data from company B's Snowflake account into company A's Snowflake account using cross-region data replication within Snowflake. Configure a direct share from company B's account to company A's account.
- C. Migrate company B's Snowflake deployment to the same region as company A's Snowflake deployment, ensuring data locality. Then perform a direct database-to-database merge of the sales data.
- D. Export the sales data from company B's Snowflake account as CSV files, and transfer the files to company A's Snowflake account. Import the data using Snowflake's data loading capabilities.

Answer: B

Explanation:
The best way to meet the requirement of consolidating company B's sales data into company A's Snowflake account is to use cross-region data replication within Snowflake. This feature allows data providers to securely share data with data consumers across different regions and cloud platforms. By replicating the sales data from company B's account in Azure West Europe region to company A's account in AWS us-east-1 region, the data will be synchronized and available for consumption. To enable data replication, the accounts must be linked and replication must be enabled by a user with the ORGADMIN role. Then, a replication group must be created and the sales database must be added to the group. Finally, a direct share must be configured from company B's account to company A's account to grant access to the replicated data. This option is more efficient and secure than exporting and importing data using CSV files or migrating the entire Snowflake deployment to another region or cloud platform. It also does not require building a custom data pipeline using external tools.
References:
* Sharing data securely across regions and cloud platforms
* Introduction to replication and failover
* Replication considerations
* Replicating account objects

NEW QUESTION # 80
A user named USER_01 needs access to create a materialized view on a schema EDW. STG_SCHEMA. How can this access be provided?

- A. GRANT CREATE MATERIALIZED VIEW ON SCHEMA EDW.STG_SCHEMA TO USER USER_01;
- B. GRANT ROLE NEW_ROLE TO USER_01;GRANT CREATE MATERIALIZED VIEW ON EDW.STG_SCHEMA

TO NEW_ROLE;

- C. GRANT ROLE NEW_ROLE TO USER USER_01;GRANT CREATE MATERIALIZED VIEW ON SCHEMA ECW.STG_SCHEKA TO NEW_ROLE;
- D. GRANT CREATE MATERIALIZED VIEW ON DATABASE EDW TO USER USERJD1;

Answer: A

Explanation:
* The correct answer is A because it grants the specific privilege to create a materialized view on the schema EDW.STG_SCHEMA to the user USER_01 directly.
* Option B is incorrect because it grants the privilege to create a materialized view on the entire database EDW, which is too broad and unnecessary. Also, there is a typo in the user name (USERJD1 instead of USER_01).
* Option C is incorrect because it grants the privilege to create a materialized view on a different schema (ECW.STG_SCHEKA instead of EDW.STG_SCHEMA). Also, there is no need to create a new role for this purpose.
* Option D is incorrect because it grants the privilege to create a materialized view on an invalid object (EDW.STG_SCHEMA is not a valid schema name, it should be EDW.STG_SCHEMA). Also, there is no need to create a new role for this purpose.
References:
* Snowflake Documentation: CREATE MATERIALIZED VIEW
* Snowflake Documentation: Working with Materialized Views
* [Snowflake Documentation: GRANT Privileges on a Schema]

NEW QUESTION # 81
......

2026 Latest ActualTorrent ARA-C01 PDF Dumps and ARA-C01 Exam Engine Free Share: https://drive.google.com/open?id=19xVAHT86HpzdkkhyuRZxtDtYyY3EHA47