Valid Test ARA-C01 Test & New ARA-C01 Exam Format

P.S. Free & New ARA-C01 dumps are available on Google Drive shared by Real4dumps: https://drive.google.com/open?id=1nQA2WUeHD5sGnXJqNU-kXYNxDqqsIyTz

The bundle has an Snowflake ARA-C01 exam questions and answers, desktop practice software, and web-based software. All the preparation products have been designed carefully with advice from hundreds of professional Snowflake certified experts. This Snowflake ARA-C01 exam questions preparation material has everything to achieve success in the SnowPro Advanced Architect Certification exam on the first attempt. The unique features of Real4dumps ARA-C01 Preparation products have been noted. The ARA-C01 pdf exam questions by Real4dumps have the most realistic Snowflake ARA-C01 exam questions. This ARA-C01 pdf covers all the ARA-C01 Exam Questions from the previous exam as well as the upcoming SnowPro Advanced Architect Certification exam. You don't need to consult different books for the Snowflake certification exam with the Real4dumps.

Snowflake ARA-C01 certification exam, also known as the SnowPro Advanced Architect Certification, is designed for professionals who want to demonstrate their expertise in the Snowflake cloud data platform. ARA-C01 exam is intended for individuals who have already earned the SnowPro Core Certification and have advanced knowledge and experience working with Snowflake. The Snowflake ARA-C01 certification exam is a comprehensive and rigorous test that covers a wide range of topics related to Snowflake architecture, design, implementation, optimization, and security.

Snowflake ARA-C01 or SnowPro Advanced Architect Certification Exam is a challenging and comprehensive test designed to evaluate the skills and knowledge of experienced architects in the use and implementation of Snowflake's cloud-based data warehousing solutions. ARA-C01 exam covers a wide range of topics, including data modeling, designing and optimizing data

warehouses, security, performance tuning, and advanced analytics. Passing ARA-C01 Exam is a significant achievement that demonstrates a high level of expertise in Snowflake architecture and design.

Snowflake ARA-C01 Certification Exam is an advanced-level exam that requires a deep understanding of Snowflake's architecture and best practices. ARA-C01 exam is designed to test an individual's ability to design and build scalable, secure, and high-performing data solutions on the Snowflake platform. ARA-C01 exam is intended for professionals who have several years of experience in data architecture and engineering and are seeking to validate their skills and demonstrate their expertise in the Snowflake ecosystem. Passing the SnowPro Advanced Architect Certification Exam can help professionals gain recognition in the industry and demonstrate their competence to potential employers.

>> Valid Test ARA-C01 Test <<

New Snowflake ARA-C01 Exam Format & Latest ARA-C01 Exam Online

We respect the private information of our customers. If you buy the ARA-C01 exam materials from us, you personal information will be protected well. Once the payment finished, we will not look the information of you, and we also won't send the junk mail to your email address. What's more, we offer you free update for 365 days for ARA-C01 Exam Dumps, so that you can get the recent information for the exam. The latest version will be automatically sent to you by our system, if you have any other questions, just contact us.

Snowflake SnowPro Advanced Architect Certification Sample Questions (Q108-Q113):

NEW QUESTION # 108
How can the Snowflake context functions be used to help determine whether a user is authorized to see data that has column-level security enforced? (Select TWO).

- A. Assign the accountadmin role to the user who is executing the object.
- B. Determine if there are ownership privileges on the masking policy that would allow the use of any function.
- C. Set masking policy conditions using is_role_in_session targeting the role in use for the current account.
- D. Set masking policy conditions using current_role targeting the role in use for the current session.
- E. Set masking policy conditions using invoker_role targeting the executing role in a SQL statement.

Answer: D,E

Explanation:
Snowflake context functions are functions that return information about the current session, user, role, warehouse, database, schema, or object. They can be used to help determine whether a user is authorized to see data that has column-level security enforced by setting masking policy conditions based on the context functions. The following context functions are relevant for column-level security:
* current_role: This function returns the name of the role in use for the current session. It can be used to set masking policy conditions that target the current session and are not affected by the execution context of the SQL statement. For example, a masking policy condition using current_role can allow or deny access to a column based on the role that the user activated in the session.
* invoker_role: This function returns the name of the executing role in a SQL statement. It can be used to set masking policy conditions that target the executing role and are affected by the execution context of the SQL statement. For example, a masking policy condition using invoker_role can allow or deny access to a column based on the role that the user specified in the SQL statement, such as using the AS ROLE clause or a stored procedure.
* is_role_in_session: This function returns TRUE if the user's current role in the session (i.e. the role returned by current_role) inherits the privileges of the specified role. It can be used to set masking policy conditions that involve role hierarchy and privilege inheritance. For example, a masking policy condition using is_role_in_session can allow or deny access to a column based on whether the user's current role is a lower privilege role in the specified role hierarchy.
The other options are not valid ways to use the Snowflake context functions for column-level security:
* Set masking policy conditions using is_role_in_session targeting the role in use for the current account.
This option is incorrect because is_role_in_session does not target the role in use for the current account, but rather the role in use for the current session. Also, the current account is not a role, but rather a logical entity that contains users, roles, warehouses, databases, and other objects.
* Determine if there are ownership privileges on the masking policy that would allow the use of any function. This option is incorrect because ownership privileges on the masking policy do not affect the use of any function, but rather the ability to create, alter, or drop the masking policy. Also, this is not a way to use the Snowflake context functions, but rather a way to check the privileges on

the masking policy object.
* Assign the accountadmin role to the user who is executing the object. This option is incorrect because assigning the accountadmin role to the user who is executing the object does not involve using the Snowflake context functions, but rather granting the highest-level role to the user. Also, this is not a recommended practice for column-level security, as it would give the user full access to all objects and data in the account, which could compromise data security and governance.
References:
* Context Functions
* Advanced Column-level Security topics
* Snowflake Data Governance: Column Level Security Overview
* Data Security Snowflake Part 2 - Column Level Security

NEW QUESTION # 109
A company has an external vendor who puts data into Google Cloud Storage. The company's Snowflake account is set up in Azure. What would be the MOST efficient way to load data from the vendor into Snowflake?

- A. Ask the vendor to create a Snowflake account, load the data into Snowflake and create a data share.
- B. Create an external stage on Google Cloud Storage and use the external table to load the data into Snowflake.
- C. Create a Snowflake Account in the Google Cloud Platform (GCP), ingest data into this account and use data replication to move the data from GCP to Azure.
- D. Copy the data from Google Cloud Storage to Azure Blob storage using external tools and load data from Blob storage to Snowflake.

Answer: B

Explanation:
The most efficient way to load data from the vendor into Snowflake is to create an external stage on Google Cloud Storage and use the external table to load the data into Snowflake (Option B). This way, you can avoid copying or moving the data across different cloud platforms, which can incur additional costs and latency. You can also leverage the external table feature to query the data directly from Google Cloud Storage without loading it into Snowflake tables, which can save storage space and improve performance. Option A is not efficient because it requires the vendor to create a Snowflake account and a data share, which can be complicated and costly. Option C is not efficient because it involves copying the data from Google Cloud Storage to Azure Blob storage using external tools, which can be slow and expensive. Option D is not efficient because it requires creating a Snowflake account in the Google Cloud Platform (GCP), ingesting data into this account, and using data replication to move the data from GCP to Azure, which can be complex and time-consuming. References: The answer can be verified from Snowflake's official documentation on external stages and external tables available on their website. Here are some relevant links:
* Using External Stages | Snowflake Documentation
* Using External Tables | Snowflake Documentation
* Loading Data from a Stage | Snowflake Documentation

NEW QUESTION # 110
How is the change of local time due to daylight savings time handled in Snowflake tasks? (Choose two.)

- A. A task scheduled in a UTC-based schedule will have no issues with the time changes.
- B. A task will move to a suspended state during the daylight savings time change.
- C. A task schedule will follow only the specified time and will fail to handle lost or duplicated hours.
- D. Task schedules can be designed to follow specified or local time zones to accommodate the time changes.
- E. A frequent task execution schedule like minutes may not cause a problem, but will affect the task history.

Answer: D,E

NEW QUESTION # 111
A DevOps team has a requirement for recovery of staging tables used in a complex set of data pipelines. The staging tables are all located in the same staging schema. One of the requirements is to have online recovery of data on a rolling 7-day basis.
After setting up the DATA_RETENTION_TIME_IN_DAYS at the database level, certain tables remain unrecoverable past 1 day.
What would cause this to occur? (Choose two.)

- A. The staging schema has not been setup for MANAGED ACCESS.

- B. The DATA_RETENTION_TIME_IN_DAYS for the staging schema has been set to 1 day.
- C. The DevOps role should be granted ALLOW_RECOVERY privilege on the staging schema.
- D. The staging tables are of the TRANSIENT type.
- E. The tables exceed the 1 TB limit for data recovery.

Answer: B,D

Explanation:
Explanation
* The DATA_RETENTION_TIME_IN_DAYS parameter controls the Time Travel retention period for an object (database, schema, or table) in Snowflake. This parameter specifies the number of days for which historical data is preserved and can be accessed using Time Travel operations (SELECT, CREATE ... CLONE, UNDROP)1.
* The requirement for recovery of staging tables on a rolling 7-day basis means that the DATA_RETENTION_TIME_IN_DAYS parameter should be set to 7 at the database level. However, this parameter can be overridden at the lower levels (schema or table) if they have a different value1.
* Therefore, one possible cause for certain tables to remain unrecoverable past 1 day is that the DATA_RETENTION_TIME_IN_DAYS for the staging schema has been set to 1 day. This would override the database level setting and limit the Time Travel retention period for all the tables in the schema to 1 day. To fix this, the parameter should be unset or set to 7 at the schema level1. Therefore, option B is correct.
* Another possible cause for certain tables to remain unrecoverable past 1 day is that the staging tables are of the TRANSIENT type. Transient tables are tables that do not have a Fail-safe period and can have a Time Travel retention period of either 0 or 1 day. Transient tables are suitable for temporary or intermediate data that can be easily reproduced or replicated2. To fix this, the tables should be created as permanent tables, which can have a Time Travel retention period of up to 90 days1. Therefore, option D is correct.
* Option A is incorrect because the MANAGED ACCESS feature is not related to the data recovery requirement. MANAGED ACCESS is a feature that allows granting access privileges to objects without explicitly granting the privileges to roles. It does not affect the Time Travel retention period or the data availability3.
* Option C is incorrect because there is no 1 TB limit for data recovery in Snowflake. The data storage size does not affect the Time Travel retention period or the data availability4.
* Option E is incorrect because there is no ALLOW_RECOVERY privilege in Snowflake. The privilege required to perform Time Travel operations is SELECT, which allows querying historical data in tables5.
References: : Understanding & Using Time Travel : Transient Tables : Managed Access : Understanding Storage Cost : Table Privileges

NEW QUESTION # 112
A healthcare company is deploying a Snowflake account that may include Personal Health Information (PHI).
The company must ensure compliance with all relevant privacy standards.
Which best practice recommendations will meet data protection and compliance requirements? (Choose three.)

- A. Rewrite SQL queries to eliminate projections of PHI data based on current_role().
- B. Use the External Tokenization feature to obfuscate sensitive data.
- C. Create Dynamic Data Masking policies and apply them to columns that contain PHI.
- D. Use, at minimum, the Business Critical edition of Snowflake.
- E. Use the Internal Tokenization feature to obfuscate sensitive data.
- F. Avoid sharing data with partner organizations.

Answer: B,C,D

Explanation:
* A healthcare company that handles PHI data must ensure compliance with relevant privacy standards, such as HIPAA, HITRUST, and GDPR. Snowflake provides several features and best practices to help customers meet their data protection and compliance requirements1.
* One best practice recommendation is to use, at minimum, the Business Critical edition of Snowflake. This edition provides the highest level of data protection and security, including end-to-end encryption with customer-managed keys, enhanced object-level security, and HIPAA and HITRUST compliance2. Therefore, option A is correct.
* Another best practice recommendation is to create Dynamic Data Masking policies and apply them to columns that contain PHI. Dynamic Data Masking is a feature that allows masking or redacting sensitive data based on the current user's role. This way, only authorized users can view the unmasked data, while others will see masked values, such as NULL, asterisks, or random characters3. Therefore, option B is correct.
* A third best practice recommendation is to use the External Tokenization feature to obfuscate sensitive data. External Tokenization

is a feature that allows replacing sensitive data with tokens that are
* generated and stored by an external service, such as Protegrity. This way, the original data is never stored or processed by Snowflake, and only authorized users can access the tokenized data through the external service4. Therefore, option D is correct.
* Option C is incorrect, because the Internal Tokenization feature is not available in Snowflake. Snowflake does not provide any native tokenization functionality, but only supports integration with external tokenization services4.
* Option E is incorrect, because rewriting SQL queries to eliminate projections of PHI data based on current_role() is not a best practice. This approach is error-prone, inefficient, and hard to maintain. A better alternative is to use Dynamic Data Masking policies, which can automatically mask data based on the user's role without modifying the queries3.
* Option F is incorrect, because avoiding sharing data with partner organizations is not a best practice.
Snowflake enables secure and governed data sharing with internal and external consumers, such as business units, customers, or partners. Data sharing does not involve copying or moving data, but only granting access privileges to the shared objects. Data sharing can also leverage Dynamic Data Masking and External Tokenization features to protect sensitive data5.
References: : Snowflake's Security & Compliance Reports : Snowflake Editions : Dynamic Data Masking : External Tokenization : Secure Data Sharing

NEW QUESTION # 113
.....