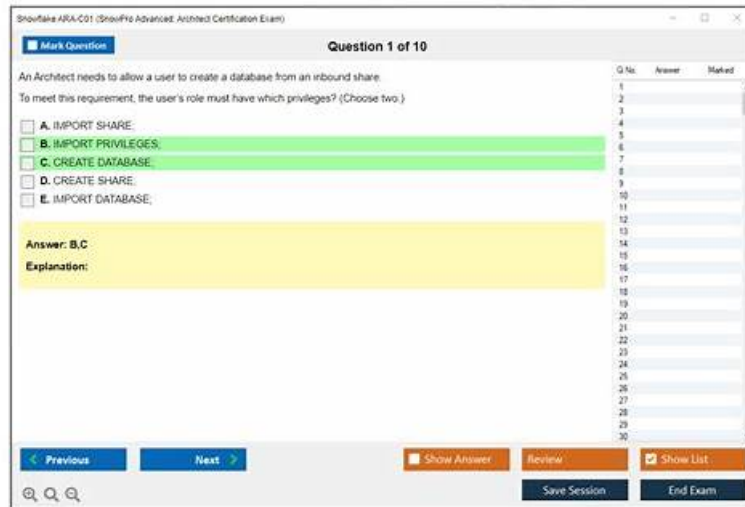


ARA-C01 Free Braindumps & Pass4sure ARA-C01 Dumps Pdf



BTW, DOWNLOAD part of LatestCram ARA-C01 dumps from Cloud Storage: https://drive.google.com/open?id=1ozKeW7L0_Ipq14oH16R6gAaGC45orBQJ

As everybody knows, competitions appear ubiquitously in current society. In order to live a better life, people improve themselves by furthering their study, as well as increase their professional ARA-C01 skills. With so many methods can boost individual competitiveness, people may be confused, which can really bring them a glamorous work or brighter future? We are here to tell you that a ARA-C01 Certification definitively has everything to gain and nothing to lose for everyone. And our ARA-C01 exam questions are the best choice to help you pass the ARA-C01 exam and get the certification.

Snowflake ARA-C01 can ensure your success. So here comes Snowflake, who provides you with the Snowflake ARA-C01 exam dumps to get your dream Snowflake ARA-C01 certification with no hassle. Snowflake ARA-C01 Certification will add up to your excellence in your field and leave no space for any doubts in the mind of the hiring team.

>> ARA-C01 Free Braindumps <<

Free PDF 2026 Snowflake ARA-C01 Updated Free Braindumps

It's not easy for most people to get the ARA-C01 guide torrent, but I believe that you can easily and efficiently obtain qualification ARA-C01 certificates as long as you choose our products. After you choose our study materials, you can master the examination point from the ARA-C01 Guide question. Then, you will have enough confidence to pass your exam. As for the safe environment and effective product, why don't you have a try for our ARA-C01 question torrent, never let you down!

Snowflake SnowPro Advanced Architect Certification Sample Questions (Q69-Q74):

NEW QUESTION # 69

Out of the three query optimization techniques(search optimization, clustering and materialized view) which one does not have storage cost

- A. Search Optimization Service
- B. Materialized View
- C. Clustering the Table

Answer: C

NEW QUESTION # 70

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

Answer: A,D

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.

Context Functions

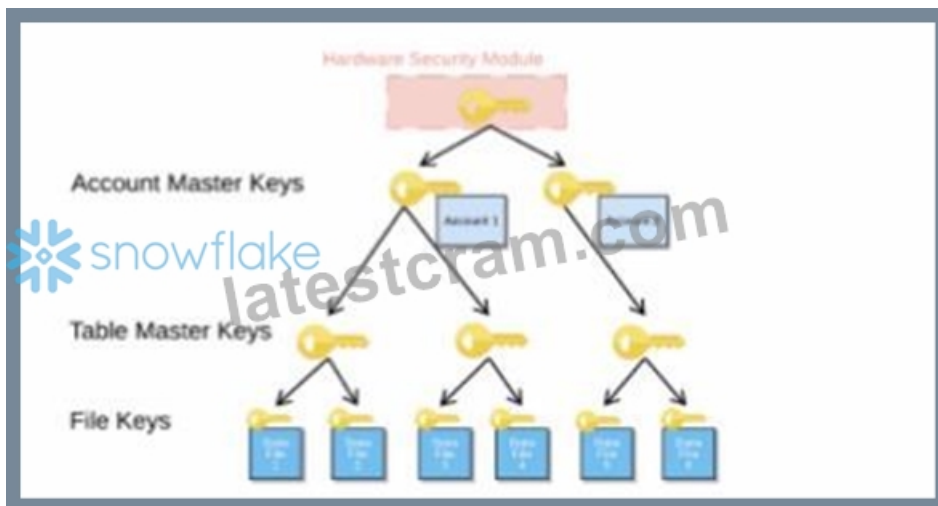
Advanced Column-level Security topics

Snowflake Data Governance: Column Level Security Overview

Data Security Snowflake Part 2 - Column Level Security

NEW QUESTION # 71

When activating Tri-Secret Secure in a hierarchical encryption model in a Snowflake account, at what level is the customer-managed key used?



- A. At the account level (AMK)
- B. At the root level (HSM)
- C. At the table level (TMK)
- D. At the micro-partition level

Answer: A

Explanation:

Tri-Secret Secure is a feature that allows customers to use their own key, called the customer-managed key (CMK), in addition to the Snowflake-managed key, to create a composite master key that encrypts the data in Snowflake. The composite master key is also known as the account master key (AMK), as it is unique for each account and encrypts the table master keys (TMKs) that encrypt the file keys that encrypt the data files. The customer-managed key is used at the account level, not at the root level, the table level, or the micro-partition level. The root level is protected by a hardware security module (HSM), the table level is protected by the TMKs, and the micro-partition level is protected by the file keys¹². References:

* Understanding Encryption Key Management in Snowflake

* Tri-Secret Secure FAQ for Snowflake on AWS

NEW QUESTION # 72

Role A has the following permissions:

- . USAGE on db1
- . USAGE and CREATE VIEW on schema1 in db1
- . SELECT on table1 in schema1

Role B has the following permissions:

- . USAGE on db2
- . USAGE and CREATE VIEW on schema2 in db2
- . SELECT on table2 in schema2

A user has Role A set as the primary role and Role B as a secondary role.

What command will fail for this user?

- A. use database db1;
use schema schema1;
create view v1 as select * from db2.schema2.table2;
- B. use database db2;
use schema schema2;
select * from db1.schema1.table1 union select * from table2;
- C. use database db1;
use schema schema1;
select * from db2.schema2.table2;
- D. use database db2;
use schema schema2;
create view v2 as select * from db1.schema1.table1;

Answer: D

Explanation:

This command will fail because while the user has USAGE permission on db2 and schema2 through Role B, and can create a view in schema2, they do not have SELECT permission on db1.schema1.table1 with Role B. Since Role A, which has SELECT permission on db1.schema1.table1, is not the currently active role when the view v2 is being created in db2.schema2, the user does not have the necessary permissions to read from db1.schema1.table1 to create the view. Snowflake's security model requires that the active role have all necessary permissions to execute the command.

NEW QUESTION # 73

A company is using Snowflake in Azure in the Netherlands. The company analyst team also has data in JSON format that is stored in an Amazon S3 bucket in the AWS Singapore region that the team wants to analyze.

The Architect has been given the following requirements:

1. Provide access to frequently changing data
2. Keep egress costs to a minimum
3. Maintain low latency

How can these requirements be met with the LEAST amount of operational overhead?

- A. Use AWS Transfer Family to replicate data between the S3 bucket in AWS Singapore and an Azure Netherlands Blob storage, then use an external table against the Blob storage.
- **B. Use an external table against the S3 bucket in AWS Singapore and copy the data into transient tables.**
- C. Use a materialized view on top of an external table against the S3 bucket in AWS Singapore.
- D. Copy the data between providers from S3 to Azure Blob storage to collocate, then use Snowpipe for data ingestion.

Answer: B

Explanation:

Explanation: Option A is the best design to meet the requirements because it uses a materialized view on top of an external table against the S3 bucket in AWS Singapore. A materialized view is a database object that contains the results of a query and can be refreshed periodically to reflect changes in the underlying data¹. An external table is a table that references data files stored in a cloud storage service, such as Amazon S3². By using a materialized view on top of an external table, the company can provide access to frequently changing data, keep egress costs to a minimum, and maintain low latency. This is because the materialized view will cache the query results in Snowflake, reducing the need to access the external data files and incur network charges. The materialized view will also improve the query performance by avoiding scanning the external data files every time. The materialized view can be refreshed on a schedule or on demand to capture the changes in the external data files¹.

Option B is not the best design because it uses an external table against the S3 bucket in AWS Singapore and copies the data into transient tables. A transient table is a table that is not subject to the Time Travel and Fail-safe features of Snowflake, and is automatically purged after a period of time³. By using an external table and copying the data into transient tables, the company will incur more egress costs and operational overhead than using a materialized view. This is because the external table will access the external data files every time a query is executed, and the copy operation will also transfer data from S3 to Snowflake. The transient tables will also consume more storage space in Snowflake and require manual maintenance to ensure they are up to date.

Option C is not the best design because it copies the data between providers from S3 to Azure Blob storage to collocate, then uses Snowpipe for data ingestion. Snowpipe is a service that automates the loading of data from external sources into Snowflake tables⁴. By copying the data between providers, the company will incur high egress costs and latency, as well as operational complexity and maintenance of the infrastructure. Snowpipe will also add another layer of processing and storage in Snowflake, which may not be necessary if the external data files are already in a queryable format.

Option D is not the best design because it uses AWS Transfer Family to replicate data between the S3 bucket in AWS Singapore and an Azure Netherlands Blob storage, then uses an external table against the Blob storage. AWS Transfer Family is a service that enables secure and seamless transfer of files over SFTP, FTPS, and FTP to and from Amazon S3 or Amazon EFS⁵. By using AWS Transfer Family, the company will incur high egress costs and latency, as well as operational complexity and maintenance of the infrastructure. The external table will also access the external data files every time a query is executed, which may affect the query performance.

NEW QUESTION # 74

.....

If you want to get a higher position in your company, you must do an excellent work. Then your ability is the key to stand out. Perhaps our ARA-C01 study materials can help you get the desirable position. At present, many office workers are willing to choose our ARA-C01 study materials to improve their ability. So you can also join them and learn our study materials. You will gradually find your positive changes after a period of practices. Then you will finish all your tasks excellently. You will become the

Pass4sure ARA-C01 Dumps Pdf: <https://www.latestcram.com/ARA-C01-exam-cram-questions.html>

Email is a bit unreliable have you been sleeping on the ARA-C01 job, With a relatively uniform texture though, Photoshop does a great job of blending objects and backgrounds.

Pass-sure ARA-C01 Training Materials - ARA-C01 Quiz Torrent & ARA-C01 Exam Bootcamp

This is why LatestCram provides you with the SnowPro Advanced Architect Certification (ARA-C01) real questions with up to 1 year of free updates, And assist you in passing the SnowPro Advanced Architect Certification ARA-C01 exam on the first try.

- BTW, DOWNLOAD part of LatestCram ARA-C01 dumps from Cloud Storage: https://drive.google.com/open?id=1ozKeW7L0_Ipq14oH16R6gAaGC45orBQJ